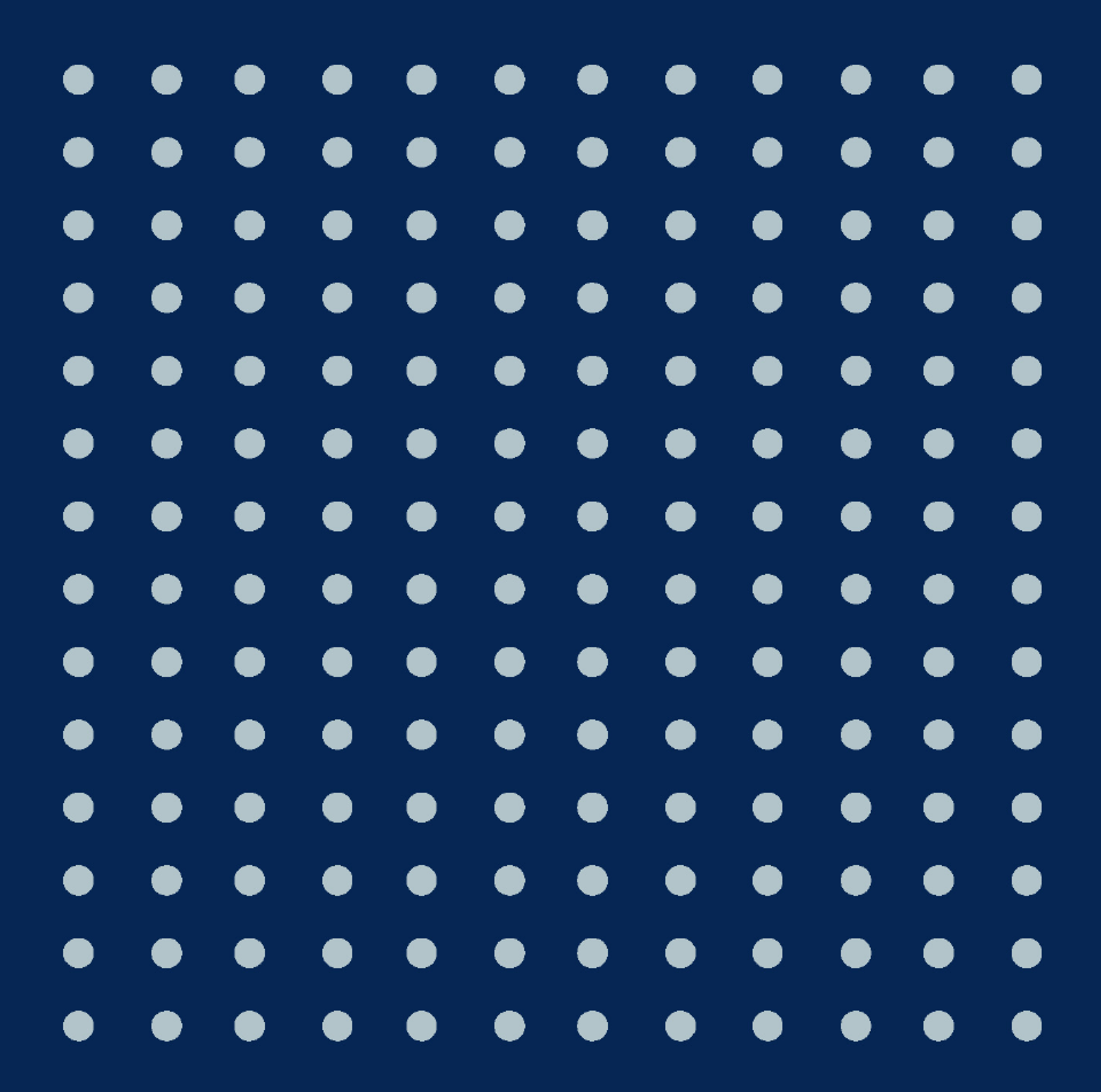
重点大学计算机专业系列教材

网络协议与网络安全（第2版）

凌 力 编著



清华大学出版社

重点大学计算机专业系列教材

网络协议与网络安全

（第 2 版）

Network Protocols & Security

（Ⅱ）

凌　力 编著

清华大学出版社
北 京

内容简介

本书由两个知识板块构成：网络协议原理和网络安全原理。两者有一定的相互独立性，可以分为两门课程讲授或学习，同时也存在较强的关联性，贯穿起来学习更有助于全面掌握网络技术，奠定扎实的理论基础。

网络协议原理部分的重点是Internet技术，其次包括Ethernet、WLAN、自组网(Ad-hoc)、宽带网络和移动通信网络等重要的网络类型及其技术。从计算机网络OSI原理出发，具体剖析了各种网络协议、网络体系结构、路由算法和多媒体信息编码算法。此外，还分析了物联网、云计算、移动计算等新技术的基本概念、技术原理和发展趋势。

在网络安全原理部分，逐一详解了古典加密算法、对称密钥加密算法、非对称密钥加密算法和单向函数加密算法以及以密码学理论为基础形成的数字签名技术、网络安全协议和密钥管理方法，并通过对网络安全威胁技术的具体分析，详细讨论了网络安全防范的技术和体系。

本书作为适合高等院校计算机、网络、通信、信息等相关专业学科的研究生和本科生教材，也可作为其他专业学生的选修、自学的参考材料。

图书在版编目(CIP)数据

网络协议与网络安全/凌力编著. --2版. --北京：清华大学出版社，2012.10
(重点大学计算机专业系列教材)
ISBN 978-7-302-28940-1

Ⅰ.①网… Ⅱ.①凌… Ⅲ.①计算机网络－通信协议－高等学校－教材 ②计算机网络－安全技术－高等学校－教材 Ⅳ.①TN915.04 ②TP393.08

中国版本图书馆CIP数据核字(2012)第110894号

责任编辑：魏江江　赵晓宁
封面设计：
责任校对：焦丽丽
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　**邮　　编：**100084
社 总 机：010-62770175　**邮　　购：**010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　**印　　张：**18.5　**字　　数：**452千字
版　　次：2012年10月第1版　**印　　次：**2012年10月第1次印刷
印　　数：1～　000
定　　价：　.00元

产品编号：036962-01

INTRODUCTION

出版说明

随着国家信息化步伐的加快和高等教育规模的扩大，社会对计算机专业人才的需求不仅体现在数量的增加上，而且体现在质量要求的提高上，培养具有研究和实践能力的高层次的计算机专业人才已成为许多重点大学计算机专业教育的主要目标。目前，我国共有16个国家重点学科、20个博士点一级学科、28个博士点二级学科集中在教育部部属重点大学，这些高校在计算机教学和科研方面具有一定优势，并且大多以国际著名大学计算机教育为参照系，具有系统完善的教学课程体系、教学实验体系、教学质量保证体系和人才培养评估体系等综合体系，形成了培养一流人才的教学和科研环境。

重点大学计算机学科的教学与科研氛围是培养一流计算机人才的基础，其中专业教材的使用和建设则是这种氛围的重要组成部分，一批具有学科方向特色优势的计算机专业教材作为各重点大学的重点建设项目成果得到肯定。为了展示和发扬各重点大学在计算机专业教育上的优势，特别是专业教材建设上的优势，同时配合各重点大学的计算机学科建设和专业课程教学需要，在教育部相关教学指导委员会专家的建议和各重点大学的大力支持下，清华大学出版社规划并出版本系列教材。本系列教材的建设旨在"汇聚学科精英、引领学科建设、培育专业英才"，同时以教材示范各重点大学的优秀教学理念、教学方法、教学手段和教学内容等。

本系列教材在规划过程中体现了如下一些基本组织原则和特点。

（1）面向学科发展的前沿，适应当前社会对计算机专业高级人才的培养需求。教材内容以基本理论为基础，反映基本理论和原理的综合应用，重视实践和应用环节。

（2）反映教学需要，促进教学发展。教材要能适应多样化的教学需要，正确把握教学内容和课程体系的改革方向。在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养，为学生知识、能力、素质协调发展创造条件。

（3）实施精品战略，突出重点，保证质量。规划教材建设的重点依然是专业基础课和专业主干课；特别注意选择并安排了一部分原来基础比较好的优秀教材或讲义修订再版，逐步形成精品教材；提倡并鼓励编写体现重点大学

计算机专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。专业基础课和专业主干课教材要配套,同一门课程可以有多本具有不同内容特点的教材。处理好教材统一性与多样化的关系;基本教材与辅助教材以及教学参考书的关系;文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优落实。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

教材编委会

FOREWORD

第2版 前言

信安山有石室，王质入其室，见二童子对弈，看之。局未终，视其所执伐薪柯已烂朽，遂归，乡里已非矣。

——东晋穆帝永和年间（公元 345 年）虞喜作《志林》

从 2007 年本书的第 1 版到 2012 年的第 2 版，时间悄悄地过去了 5 年。5 年不算久，弹指一挥间，够一名学子读完大学。但网络这个虚拟世界或许有自己的历法，颇有“洞中方七日，世上已千年”的“仙界”之风，只不过颠倒过来，日子过得飞快，现实世界的 5 年，互联网恰似飞越了 50 年，否则无法解释为何短时间内会冒出那么多新技术，产生那么巨大的变化。

比起那位在山中看仙童下棋的樵夫，我们不知道是幸或不幸。可怜的砍柴大哥仅仅偶发雅兴，旁观了一盘棋，结果连斧柄都朽烂了，自己的家和村子竟然物非人亦非。而我们 5 年前的 PC 和手机呢？朽烂倒未必，可是十有八九已被我们自己视为“破烂”，不少人的手机也许已经换了几部。

当然可以坚持不换老的 PC，但用 PC 上网意味着必须端端正正地坐在桌子前，不像用平板计算机一样自由自在，不能走着看新闻，躺着看视频，或在地铁里悠闲地边听音乐边读书；用 PC 还意味着只能老老实实地用鼠标点击，而无法划拉、抓挠屏幕，与蔬菜、小鸟和怪兽作斗争。假如依依不舍，钟爱多年的老手机也不一定换。但如今，堂堂的嵌入式操作系统居然被打上了“非智能”的标签，成为“低能”一族，举目皆是多点触控的大屏智能机，桌面上各式各样的 App（应用程序）浮动着，竞相争宠。你实现 3G 上网，我就有 WiFi 热点互联；你可以 GPS（全球定位系统）导航，我就提供 LBS（位置相关服务）业务。因此，PC 和手机换与不换，这是一个问题。

这一切主要是拜乔布斯所赐。他的苹果既不是来自亚当和夏娃的伊甸园，也不是摘自牛顿头上的苹果树，而是被图林（计算机与人工智能之父）咬过的那一颗，半个多世纪后，完全退去了毒性，剩下的只有熠熠闪光的智慧、创新和维生素。刚刚过去的 5 年，打着苹果标志的平板计算机和手机覆盖了全球，更重要的是这些漂亮非凡、人见人爱的“尤物”改变了人们对计算机的认知、改变了一成不变的计算机操作习惯，让上网变成一种令人愉悦的旅程，

让获取信息像呼吸一样简单和自然,让地球上的任意两个人的沟通仿佛面对面。

所以,我们有理由相信,虚拟世界的时空超越了爱因斯坦的学说,时钟走得飞快,日子变得很短,距离趋近于零,而虚拟世界的演进速度则超过了达尔文的想象,甚至智能“生物”的诞生也并非痴人说梦,或许为期不远了。

5 年来,山洞中的棋局大概还没走出一步,恰似 IPv4 协议依旧,Internet 仍然不安全,国足还是没出线,但尘世间一直不可遏止地忙碌着,不论是现实世界还是虚拟世界,沧海总是或快或慢地往桑田方向变。想想云计算,想想物联网吧,早上晴空万里,下午就“云雾”缭绕了。闲不住的乔布斯驾鹤西去后肯定会带去人类的进取精神,这样仙童下棋的石桌就有机会支持滑屏了。

祝大家享受知识,也享受知识创造的成果。

编　者

2012 年 4 月于复旦大学

FOREWORD

第1版 前言

■ 关于观点

我们不要惧怕亮出自己的观点——即使有失偏颇也可供大家评判、批评、争论、思考、品味。客观的数据是必要的，枯燥的理论是难免的，但最终还是要有观点和结论，这关系到立场问题。

在本书中可以找到许多观点，谦虚地说，它们不一定都是正确的。影响正确性判断的原因有三：一是认识有偏颇；二是事物在发展，丑小鸭长啊长，说不定是只白天鹅呢！三是有更多更好的事物在孕育中、诞生中。这正是计算机网络领域的特点：朝气蓬勃，欣欣向荣，这也是研究计算机网络的乐趣所在：推陈出新，创意无限，不过这也是置身计算机网络行业的辛苦所在：三日不见，形同陌路！

理解是观点的起步，观点是理解的体现。理解了，有观点了，往往就成功了一半。

■ 关于情感

为什么大数学家说在天书般的方程式堆中漫游是妙不可言、富有乐趣的？那就是情感。固然他们到达了某种常人无法企及的境界，但不可否认，任何事物都是可以有情感的，不管这种情感是其自身所具备的，还是感怀的人所赋予的。

所以我们可以大胆地说，技术文献是有情感的，论文讲义是有情感的，计算机网络也是有情感的。

计算机网络本来就是很拟人化的东西，人类逐渐在互联网上构筑起一个虚拟世界。这是一个多么富有想象力的全人类共同参与的大工程！所以，本书读者们在阅读到那些看上去和计算机网络术语、技术无关的文字时，请不要忽略，相信我，它们是属于计算机网络领域的。

■ 关于角度

但凡足球运动员用大家认为不可思议的角度射入一个球时，都会得到暴风雨般的喝彩，因为这个球与众不同，因为把自己从昏昏欲睡的比赛过程中

弄醒了。没有统计过,不知道有多少人在阅读技术文章时处于半清醒状态,我想应该不在少数,因为不客气地说,大部分的教科书都很教科书,有千篇一律的公式和行文,有从基础到深入的严密逻辑。

然而我不认为这有什么可取的,或者说不认为一定非得这样。花开两朵还各有不同呢!本书抓的一个个的点并不全落在“网”的“纲”上,但希望抓起这些点能有一些“目”能够张开,由点及面,窥一斑知全豹,也是一个了解“全网”的不容易睡着的方法。

本书的每一个章节都试图去了解一个或大或小的问题,并扩展到周围的问题。我们比较多地采用“比较性分析”的办法,通过对比来加深概念和技术方法的理解。不过是否能达到预期效果——还得靠读者自己琢磨!

■ 关于细节

有人说“细节决定成败”,那大概是在说装配钟表吧。回想童年,你还记得多少细节?——无非是和谁闹别扭了,后来又要好了等等,至于为什么事情吵架多半已经不记得了。不要慌张,不是健忘,那是人脑的优秀性能之一:忘记不必要的东西。那样你才没有崩溃。

计算机网络中也有很多细节,它们是由许许多多数字、字母构成的,足够把人弄疯。所以,我们最好要学会什么时候要关注细节(当少尉排长),什么时候要看重全局(当五星将军)。

普天下阐述技术细节的书已经够多了,所以本书不打算凑这个热闹。而当我们把一些所谓的细节屏蔽掉之后,我们往往会发现技术的核心思想显露出来了。至于细节,很简单,我们可以检索有关资料来获取。

■ 关于本教材

- 本教材可作为本科生、研究生课程教学用书或参考书。
- 虽然涉及“网络协议”,但并非以单纯的网络协议原理为首要内容,而是从较为宏观的网络与通信技术的角度来体现较为微观的协议技术,旨在从网络理解协议、从协议透视网络。
- 每章为一个大的知识点,阐述一个网络及其安全技术的“板块”;每个板块由若干小的知识点构成。各板块间的内容略有交叉重叠,反映出网络通信领域各项关键技术的关联性,因此,既要对各个“分支”进行深入钻研,又要有全局的、整体的观念。
- 每章学习约需一两个“单元时间”(每单元时间为两三课时)。
- 由于涉及的概念、术语较多,这个“基础”打得不会很轻松,但会从知识面拓展上有所获益。
- 希望在学习过程中不要仅仅局限于书本内容,而应该主动去学习更多的相关知识,对感兴趣的问题予以纵深挖掘。这里包含两层意思:第一,鼓励借助计算机、互联网进行学习,而不要停留在书本涵盖的有限知识上;第二,把对理论的理性认识应用于实践中,获得感性认识,达到融会贯通的目的。
- 勤于思考,不迷信“权威”。对“可疑”观点应勇敢地提出质疑和自己的见解。同时也要善于发现问题、提出问题,并提高分析问题、解决问题的能力。
- 因为网络技术的发展速度很快,本教材也将不断更新,力争与本领域的新进展同步。

作　者

2007 年 6 月于复旦大学

CONTENTS

目录

CHAPTER 1

第1章 计算机网络与协议

网络在自然界广泛存在,例如,由江河湖海构成的水系,蜘蛛辛勤编织的捕食网。网络在人类社会里更是无处不在,简单到渔家的渔网、猎人的捕兽网,复杂到遍布城市和乡村的道路网、输电网、输气网、自来水管网、下水道网,乃至错综复杂的人际关系网、国际关系网、疏而不漏的法网。网络无疑是人类赖以生存的重要手段之一。

网络体现了人类的大智慧。通过组网,可以建立相互间的联系,可以远程输送资源,可以实现 1+1>2 的集聚和放大效应。

网络的关键是通过经纬脉络组成某种结构,进而产生特定功效。因此,由空洞起作用的筛子不能算是网络,而同样被编织成网形的渔网则是一种网络。

英语中有许多词汇指代网络,除了 Network 和 Net,还有 Grid、Web、Matrix、Mesh 等,含义上有细微的差别,在网络技术术语中经常会出现。

网络造就了我们身处的信息时代,并被赋予了更为丰富的内涵。电话网、广播电视网、计算机网,满足了人们及时掌握信息、方便沟通交流、实现资源共享的需要,于是,越来越多的人不知不觉中对网络有了严重的、几乎不可逆转的依赖性,因为网络已经深入到了学习、工作和日常生活的方方面面。且不论这种依赖性本身孰是孰非,网络给人们带来的高效性、便捷化、无疆界是有目共睹的。仅凭这一点,网络无疑担当了这个时代当仁不让的主角。

人们平时说的“网络”,通常就是指计算机网络,甚至就是指 Internet。这个现象很令人深思,但并不费解。原因在于计算机和计算机网络引发了汹涌澎湃的数字化浪潮,不仅把传统的模拟网络逐一纳入怀中成为数字网络,而且通过替代或融合,把各种类型的计算机网络逐步统一为 Internet。或许有人会争论 Internet 一家独大的局面是否存在弊端的问题,但越来越多的人开始或已经习惯于上网搜信息、发言论、找朋友、买商品,谁会希望倒退到没有 Internet 的年代?计算机网络和 Internet 正是本书的重点。

1946 年,第一台计算机问世;1969 年,计算机网络诞生。计算机具备与生俱来的运算能力,可以让人摆脱烦琐的计算任务,让运算速度得到以数量

级计的提高。计算机除了能够解决数学问题,还能够记录和处理文字、图像等各类信息,并可开发出分析、推断能力,为创立人类大脑以外的第二种“智能体”奠定了扎实的基础、开辟出巨大的想象空间。没有计算机,当然就没有计算机网络;但是反过来,没有计算机网络,计算机的发展也会受到极大的制约。一台孤立的计算机能完成的任务是相当有限的,并且无法实现硬件、软件、信息资源共享而使价值最大化。短短几十年间,计算机从价格昂贵的庞然大物,到人们口袋中的随身物品,而且可以随时联网,正是计算机和计算机网络技术相互促进、共同发展的结果。我们乐见这样的结果,而且希望不久的将来就会实现无处不在的普适计算和无须刻意为之的自然计算。

计算机网络(Computer Network)就是计算机设备通过物理媒介和通信协议的互连构成的信息传输系统。

计算机设备可以是主机、服务器、个人计算机、智能手机、交换机、路由器、网关等任意一种终端或联网设施;物理媒介可以是有线或无线信道;通信**协议**(protocol)则是计算机设备间相互对话的特定机制,实现信息的无差错传输。

1.1 计算机网络分类

网络技术类型众多。从不同的角度,可以把网络划分为不同类别。

本书主要考察以信息传输为目的的网络,但不妨把过于古老的通信方式排除在外,如烽火台、信鸽、驿站、邮政信箱等。这样,网络大致可分为模拟网络和数字网络两种类型。前者采用模拟传输技术,包括传统的电报网、电话网、广播(电台)网、电视网,一般存在信道容量小、传输质量差、抗干扰能力弱、安全性低等问题;后者采用数字传输技术,包括计算机网络、数字电视网、数字广播网、网络电话、卫星网,其共同点是能够传输数字化的多媒体信息,具备高效率、高可靠的特点,而且具有很强的扩展潜力,可以支持双向互动和个性访问,是网络技术的发展方向。新一代的电话网、有线电视网实际上都采用了计算机网络技术,为实现“三网融合”扫清了技术障碍。

可见,数字化是信息传输技术的分水岭。例如,第一代移动电话系统属于模拟网络,从第二代开始,就进化为数字网络。而数字传输技术的起源和制高点都属于计算机网络,可以说,掌握了计算机网络技术,就不难理解其他数字网络技术了。在讨论计算机网络的技术细节之前,不妨先从其不同类型入手,了解计算机网络的概貌。

计算机网络可以依据其各种属性进行如下分类。

(1) 按通信媒介分为有线网络、无线网络。移动电话网、卫星网也属于无线网络范畴,但需要注意的是,无线网络并非必须完全由无线信道组成。

(2) 按覆盖范围分为**广域网**(Wide Area Network,WAN)(覆盖国家到全球)、**城域网**(Metro-politan Area Network,MAN)(通常覆盖一个城市)、**局域网**(Local Area Network,LAN)(可以覆盖小到家庭大到校园的范围)、**个域网**(Personal Area Network,PAN)(主要指桌面或人体的覆盖)。不同的网络覆盖范围没有严格的界限,相互间可以交叉重叠。

(3) 按拓扑结构(如图 1.1 所示),分为线型(包括总线型和串联型)、环型、星型、树型和网型。不同拓扑结构的网络不仅是互联形式上的差异,性能上也可能存在很大差别。例如,线型网络存在单点故障,而与之近似的环型网络就不存在单点故障;星型网络可以转化为

两层结构的树型网络；星型网络比较适合于构造卫星网，树型网络则常用于层次性管理体系。此外，在给定的物理拓扑结构上，可以通过协议控制，形成不同的逻辑拓扑结构。

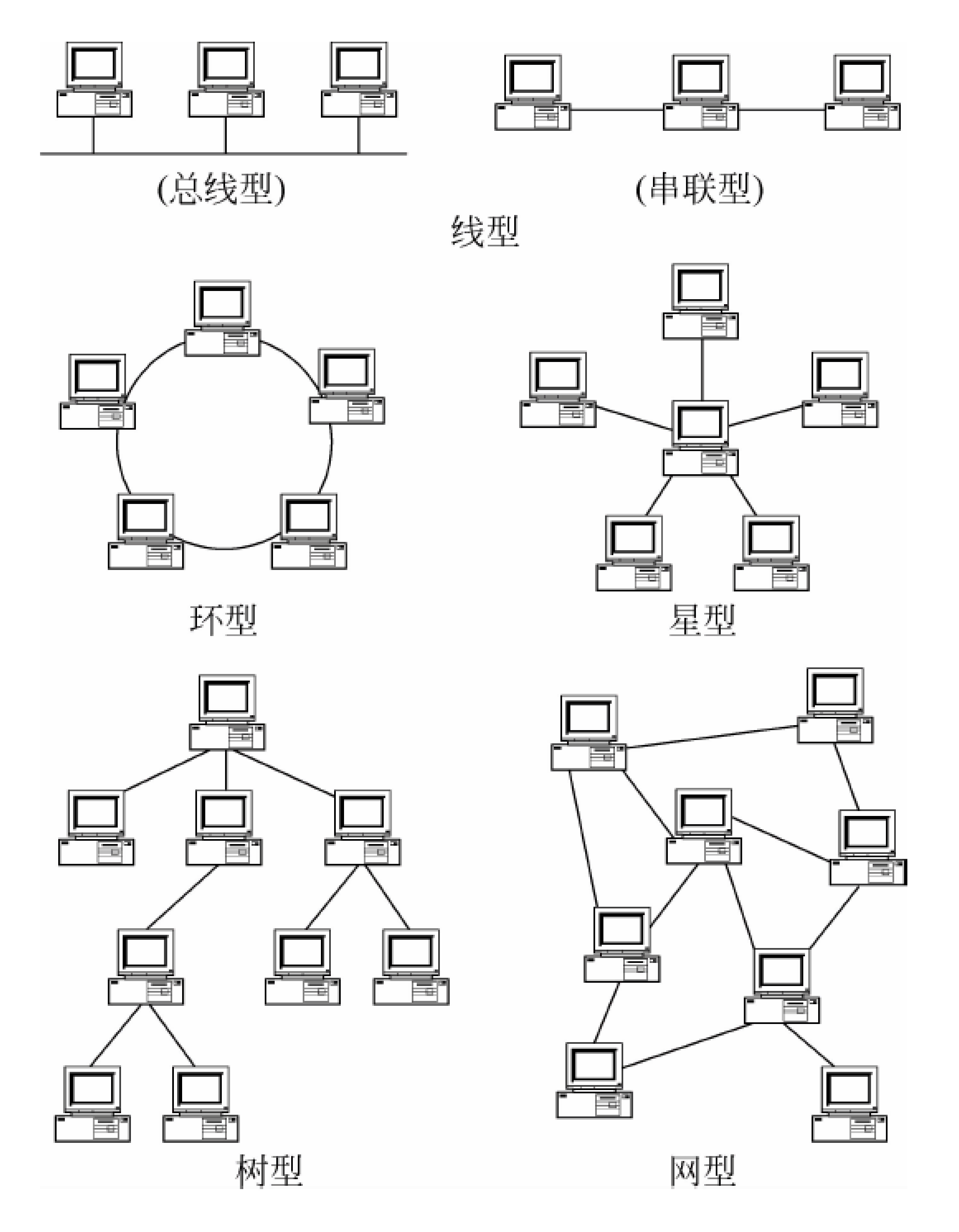


图 1.1　网络拓扑结构分类图示

1.2　开放系统互连模型

1.2.1　网络协议标准化

研究计算机网络，应从其开放性(opening)的本质入手。

倘若听任计算机网络技术自由发挥而不加以约束，势必会造成各自为政的不利局面，与计算机网络互联互通的本意背道而驰，因为封闭的、相互隔离的网络是没有太大实际意义的。对于采用不同技术的网络，即使想要实现网络互联互通，代价也会非常高昂。所以，计算机网络技术的开放性十分必要，而网络技术的标准化工作是保障其开放性的关键手段之一。

由于网络技术具有很强的全局性，因此网络领域更注重国际标准的制订，以便被全球各国普遍采纳，有利于网络互联的实现。

在网络技术文献中，我们时常会为应该使用“互连”还是“互联”犯难。实际上，对应英语单词都是 Interconnection，两者意思大致相同，但互连倾向于描述物理上的连接关系，互联则侧重逻辑上的联系，因此，网络设备之间或网际的通信更适合用“互连”一词来表达。

计算机网络的标准化与“消灭诸侯、统一国家”完全不同。假如把 OSI(开放系统互连)参考模型理解为网络技术世界的宪法，其他技术标准理解为法律、法规，那么，在 OSI 框架指导下，就可以衍生出不同的协议和技术，并制订相应的标准。

考察网络协议的标准化进程,可以发现,大部分的网络协议都是先有实践、再有标准。通过实际检验后证明比较成熟的技术,标准化组织就会“拿过来”进行完善,使其上升为通行的标准(或建议),为所有开发者所遵守。此外,有许多国际标准的形成是参照了多种企业标准或结合了不同区域性标准化组织制定的标准。

OSI技术的开放性意味着:

(1) 所有技术要求和细节都是公开发布的,任何人都有权获取并参照实现;

(2) 研究者和开发者都可提供自己的创意、改进方法,其建议有可能被标准化组织采纳,用于完善标准;

(3) 符合技术标准的系统可以接入已有系统。

正是由于计算机网络具有开放性,因此从这个意义上说,Internet一网独大其实并没有那么可怕。Internet技术标准可以为任何人所用,用于研发产品,不存在知识产权壁垒,不会形成技术垄断。

网络领域的技术标准主要由以下几个标准化组织负责研究和制订。

国际标准化组织(International Organization for Standardization,ISO),成立于1947年2月23日,其成员由来自世界上100多个国家的国家级标准化团体组成。ISO一词来源于希腊语,意为相同、平等,与标准化的目的和内涵非常吻合。

国际电工委员会(International Electrotechnical Commission,IEC),成立于1906年,于1947年并入ISO。原美国标准ASCII字符集的定义,即由ISO和IEC共同制订为国际通用标准,标准号为ISO/IEC 646。

国际电信联盟(International Telecommunications Union,ITU),是联合国的一个专门机构,设立于1865年,致力于制定国际电信标准。ITU由三部分组成:ITU-R负责无线通信(在世界范围分配无线频率);ITU- T负责制定电信标准,1956—1993年称为**国际电报电话咨询委员会**(Consultative Committee for International Telegraph and Telephone,CCITT),下设多个研究组(SG),研究组下设不同专题,例如Q42/SG VII专门研究OSI参考模型;ITU-D负责开发事务。

电气和电子工程师协会(Institute of Electrical and Electronics Engineers,IEEE),于1963年1月1日由美国无线电工程师协会(IRE,1912年成立)和美国电气工程师协会(AIEE,1884年成立)合并而成,设37个专业分会(society)和3个联合会(council),下设380多个学组(chapter)。在其制订的标准中,IEEE 802为局域网和城域网系列标准,IEEE 802.11为无线网络系列标准。

IETF(Internet Engineering Task Force),从属于Internet体系结构委员会(Internet Architecture Board,IAB),创建于1992年6月,是一个开放的国际标准化组织,由网络组织者、运营者、研究组、制造商、开发者组成,专门负责研究有关Internet的协议及其标准,以RFC(Requests For Comments)文件形式发布,例如IP由RFC 791规定。

不应该把国际标准理解为强势和强硬的死规定,实际上,技术和产品的研发者在遵循技术标准的大原则下,可以根据自身需要进行灵活的、有针对性的调整,如分层模型的改变(参见TCP/IP分层结构)、层结构的改变(如子层、交叉层)、协议补丁、不同版本的网络协议标准,也允许存在网络协议实现上的不同,如选择实现不同的功能子集、采用不同的选项(options)、自定义字段、各种配置参数等。

因此，即便参照了同一个国际标准，由于可能存在以上差异，两者仍然有可能无法完成互连。**协议一致性测试**（Protocol Conformance Testing，PCT）工作可以有助于不同系统的互连，然而，PCT 既非网络互连的充分条件，也非必要条件。

另外，考察不同标准化组织的历史轨迹和工作内容，可以发现一个有意思的现象，就是在技术思想和方法上逐步走向融合。例如，ITU 专注于电信技术，IETF 的专业领域是 Internet，如今，电信网络要向下一代网络（Next Generation Network，NGN）发展，Internet 则向着下一代互联网（Next Generation Internet，NGI）发展，两者都采用了 IPv6 协议标准，并重点发展视频、语音等多媒体通信技术。换言之，ITU 需要借助 Internet 的数字交换平台，IETF 则试图拓展电信类业务、提供电信级服务。从标准制定上也可以看到：ITU-T H. 248 与 IETF MGCP 标准相似，都是媒体网关控制协议；ITU-T H. 323 与 IETF SIP 标准类似，都是会话发起协议。

1.2.2　OSI 模型

早在 1977 年，ISO 就开始着手计算机网络的标准化工作，并于 1984 年颁布了**开放系统互连**（Opening System Interconnection，OSI）基本参考模型（Basic Reference Model），简称 OSI′ISO，用于规范网络的体系结构、通信协议和网络互连技术，从此，计算机网络进入了有序的发展阶段。

根据 OSI 基本参考模型，计算机网络中的计算机设备被划分为两个组成部分（如图 1.2 所示）。

（1）**端开放系统**（End System），即计算机终端（PC/便携电脑/智能手机）、服务器、工作站、主机等，位于网络的边缘，供用户直接操作或进行访问。网络中的内容都存放在端开放系统中。

（2）**中继开放系统**（Relay System），即交换机、路由器、网关、网桥、代理、接入服务器、基站等，位于网络内部，是实现网络联网、提供网络接入的设备。

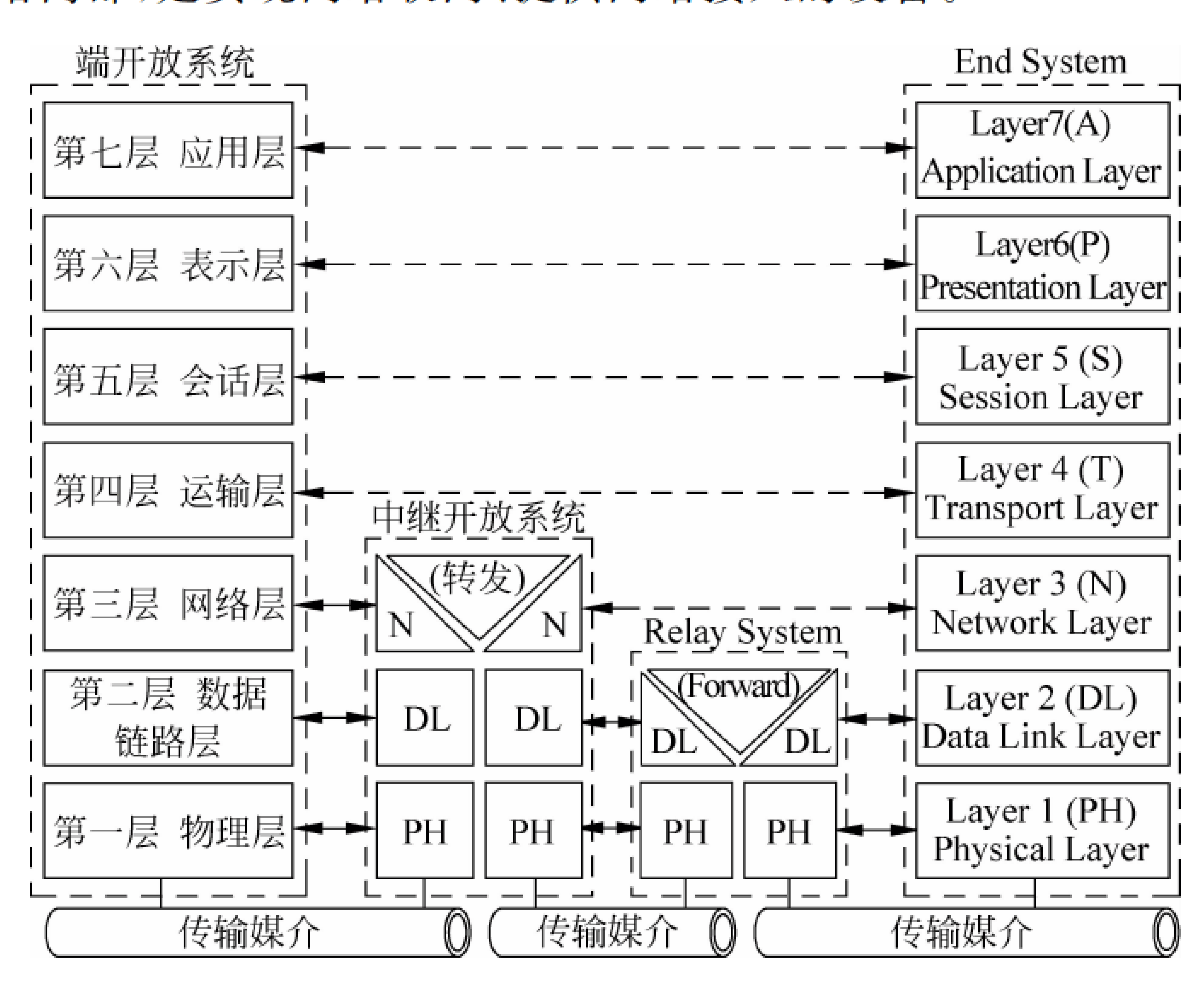


图 1.2　OSI 基本参考模型

端开放系统是数据传输的发起者(发送者)或接收者,即数据源(source)或宿(destination),是用户数据的处理者,中继开放系统只负责数据的转发工作,尽力将数据送达指定目的地,并不关心数据内容。因此,我们通常简单化地把中继开放系统和传输媒介构成的体系统称为网络,端开放系统则是接入网络的各种类型计算机设备。

我们习惯于把中继开放系统设备称为网络中的结点(node)。而 node 还可以译为节点,意思类似,且在各种文献中两者都有使用。但鉴于两者存在微小的差别,在本书中将统一采用“结点”来指代网络中的设备,而用“节点”表示某些操作流程及树状结构中的各个有序的步骤和环节。

传统的电话网(POTS)构成了覆盖全球的庞大网络体系,而且程控电话交换机的核心就是计算机,实现电话呼叫管理和网络管理的信令系统就是协议,早期的用户拨号上网也是通过电话网,但是,传统电话网却不属于计算机网络。主要原因是电话网终端(即电话机)并非智能设备(即计算机设备),不能与网络设备(电话交换机)实现对等的协议操作;其次,关键业务语音通话是通过物理电路转接实现的,采用模拟通信方式,是一种电路交换技术。

1.2.3 OSI 分层结构

OSI 基本参考模型(如图 1.2 所示)将构成计算机网络的通信实体分为七层(7 layers),层与层之间的关系体现为如下几点。

(1) 为分组交换的目的服务。与**分组交换**(Packet Switching)方式相对应的有**报文交换**(Datagram Switching)方式,后者协议比较简单,一般不进行分层;而分组交换和报文交换都属于**数据交换**(Data Switching)范畴,采用数字通信技术;与数据交换相对的交换技术则是**电路交换**(Circuit Switching),如传统电话网,采用模拟通信技术。

(2) 每个层均实现其特定的网络互连功能,可以是面向连接的,也可以是面向非连接的。

(3) 每个层都与且只与对等的层进行通信。

(4) 同一实体中,相邻的层之间是服务和被服务的关系,其中,下层为服务提供者,上层为服务使用者,包括建立连接、传输数据、拆除连接等服务功能。

(5) 层与层之间的服务调用和提供一般使用服务**原语**(primitive)、通过**服务访问点**(Service Access Point,SAP)接口来完成,典型的 SAP 方式如**套接字**(socket)接口,可创建、使用和维护 TCP/UDP 连接。

由于开放系统中不同的网络设备具有不同的层次结构和层次数,因此,一个层与其**对等层**(peer)的通信就存在不同的类型。

(1) **点对点**(Point-to-Point)通信:该层所在的网络设备是相邻的,即通过物理媒介直接连接,这种情况下,对等层之间的通信为点对点方式。例如 PH 层之间、DL 层之间的互连。

(2) **端到端**(End-to-End)通信:指端开放系统所特有的层之间的通信方式。可以理解为跨越中继网络的对等层互连。例如 T 层之间、A 层之间的互连。

但是,中继开放系统的两个对等层之间,或中继开放系统的某个层与端开放系统的对等层之间,如果需要穿越其他中继开放系统进行互连,则既不属于点对点方式,也不属于端到端方式。另外,如果把网络设备看成一个整体,那么,互连的设备间的相互关系也可使用上

述分类和表述方法。

需要注意的是，在 Internet 应用中有一类 P2P 技术，实际上是 Peer-to-Peer 的简称，而非点对点通信的缩写。另有少数文献将 P2P 称为“端对端”，其实并不准确，使用“对等通信”或“对等网络”更为规范。

OSI 分层技术使不同层次的协议承担特定的功能，有利于根据不同需求设计不同的协议，有利于实现对等层互连；可按需调用不同的低层协议的服务，如有线网络或无线网络，形成协议栈的灵活构造；一个层次协议的改变不会牵一发动全身，便于网络系统实施改进、升级。

OSI 基本参考模型取之于网络、用之于网络。OSI 总结各种网络互连技术并予以标准化，对于网络协议和设备的研究、创新、开发均具有很强的指导意义。现有的网络技术均遵循 OSI 技术框架。

OSI 各层所承担的互连及其服务功能如下。

(1) 第一层为**物理层**(Physical Layer)。物理层用于驱动物理媒介 I/O，实现数据发送和接收，必要时进行信道的载波侦听、调制和解调、编码和解码。

(2) 第二层为**数据链路层**(Data Link Layer)。数据链路层利用物理信道(物理层)实现透明的数据传输，并向上层提供虚拟的数据传输链路。数据传输可以是可靠的，也可以不保证可靠性。在共享信道上，数据链路层还可根据地址进行数据报文的选择性接收。

数据链路层的协议数据单元通常被称为**帧**(frame)。

思考：*为什么允许一个层的数据传输服务可以不保证可靠？可以采用哪些技术手段保障可靠数据传输？*

(3) 第三层为**网络层**(Network Layer)。

网络层主要提供寻址、路由功能，并按照数据链路的要求进行分片和重组。网络层的协议数据单元通常称为**分组**(packet)，故网络层常被称为**分组层**(Packet Layer)。

分组层可以提供**虚电路**(Virtual Circuit，VC)或称**逻辑电路**(Logical Circuit，LC)的业务，可复用数据链路，具体可采用以下四种类型之一。

① **交换型虚电路**(Switching VC，SVC)需要通过虚电路连接建立过程来创建，使用完毕后释放该虚电路。SVC 支持双向数据传输，由于可按需维护连接，因此灵活性很强。

② **永久性虚电路**(Permanent VC，PVC)不需要连接建立过程，而是通过配置(由人工或信令协议控制完成)来创建虚电路，可直接使用连接标识来收发数据，提高了数据传输效率。

③ **单向入虚电路**(Incoming VC，IVC)只接收入站数据，而不进行出站数据发送。单向入虚电路的这种性质往往是人为限定的，所以是一种逻辑上的单向性。

④ **单向出虚电路**(Outgoing VC，OVC)与 IVC 类似，但只完成出站数据发送任务。

(4) 第四层为**运输层**(Transport Layer)。运输层的主要任务是提供端到端的数据传输。这是由运输层所处的特殊位置决定的，因为运输层通常是跨越网络互连的最低一层，所以，假如实现了运输层之间的对等通信，就意味着端系统之间(源和宿)已经成功达成了互连关系，尽管这种互连关系可能是面向连接的、可靠的，也可能是面向非连接的、不保证可靠的。运输层还可以承担数据链路或虚连接的复用/解复用、分流/合流。

(5) 第五层为**会话层**(Session Layer)。会话层的核心功能是为网络应用维持所需的会

话关系。如图 1.3 所示,一个应用(应用活动)可以只需要一个会话就完成,也可以由多个阶段的多次会话串联起来构成,还可以由多个并列的会话来构成,但一个会话通常不会被多个应用所共享(思考为什么)。

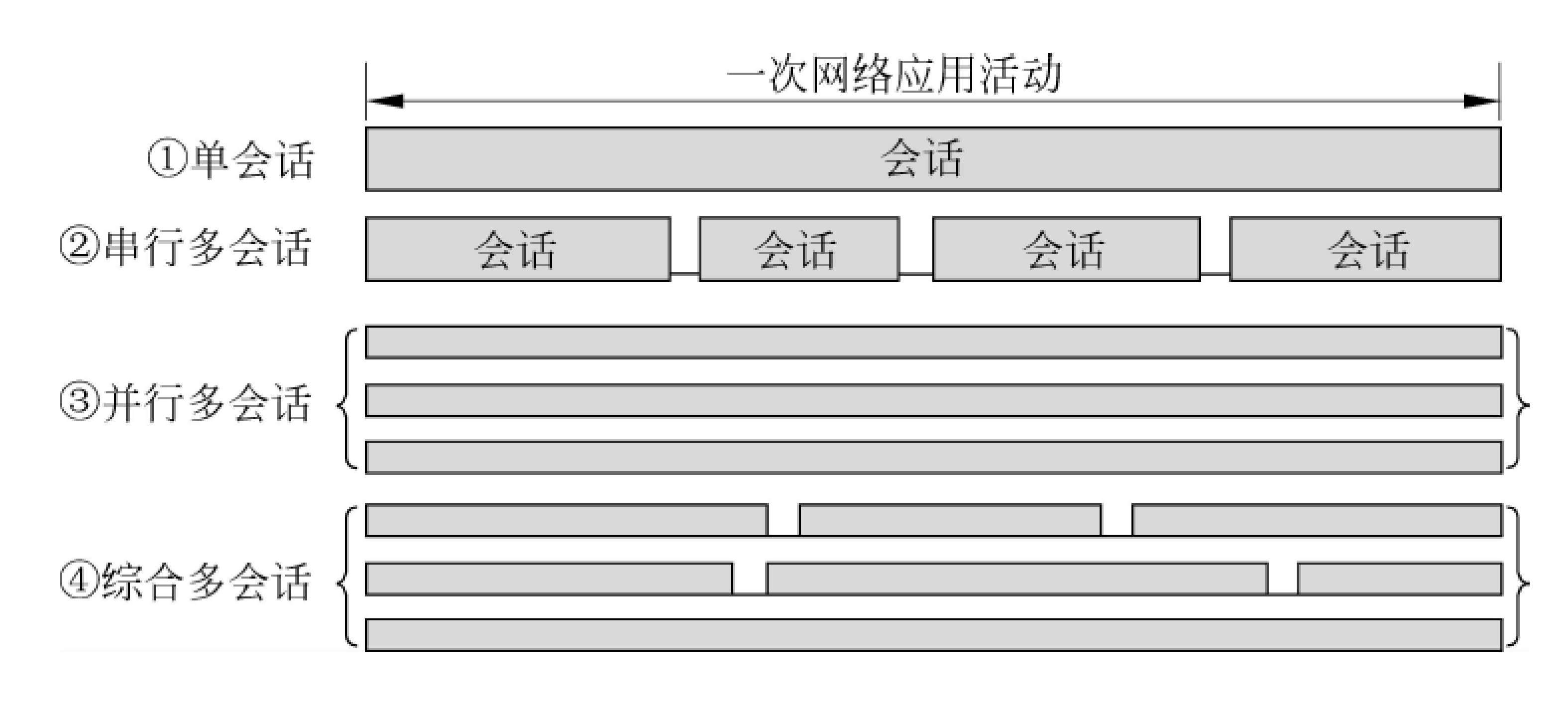


图 1.3　网络应用活动与会话关系示意

例如,一项应用任务需要从数据库服务器 A 和 B 上分别获取数据,然后根据计算结果从文件服务器 C 上下载文件,接着转发给用户 P_1、P_2 和 P_3,最后将相关数据写入服务器 D。使用会话管理的机制可以相对独立地实现复杂应用中各个应用子活动,并可灵活地组合、容易扩展和维护。

(6) 第六层为**表示层**(Presentation Layer)。计算机平台和应用系统的编码方式可能互不相同,为了实现互连,编码 $C_1, C_2, \cdots, C_n$ 就需要相互转换。如图 1.4(a)采用了两两互转方法,每台设备需要 $n-1$ 个转换模块,每加入一个新设备(新编码 C_k)不仅要实现 n 个转换模块,而且已有的设备都要受到影响,需增加一个到新编码的转换模块;如图 1.4(b)所示则设计了一种不依赖于设备的标准编码 C_0,所有设备一律只实现到 C_0 的转换,互转即 $C_i \to C_0 \to C_j$,那么,每台设备始终只需具备一种转换能力,增加新设备也不会干扰现有设备。显然,第二种方案具有更大优势。

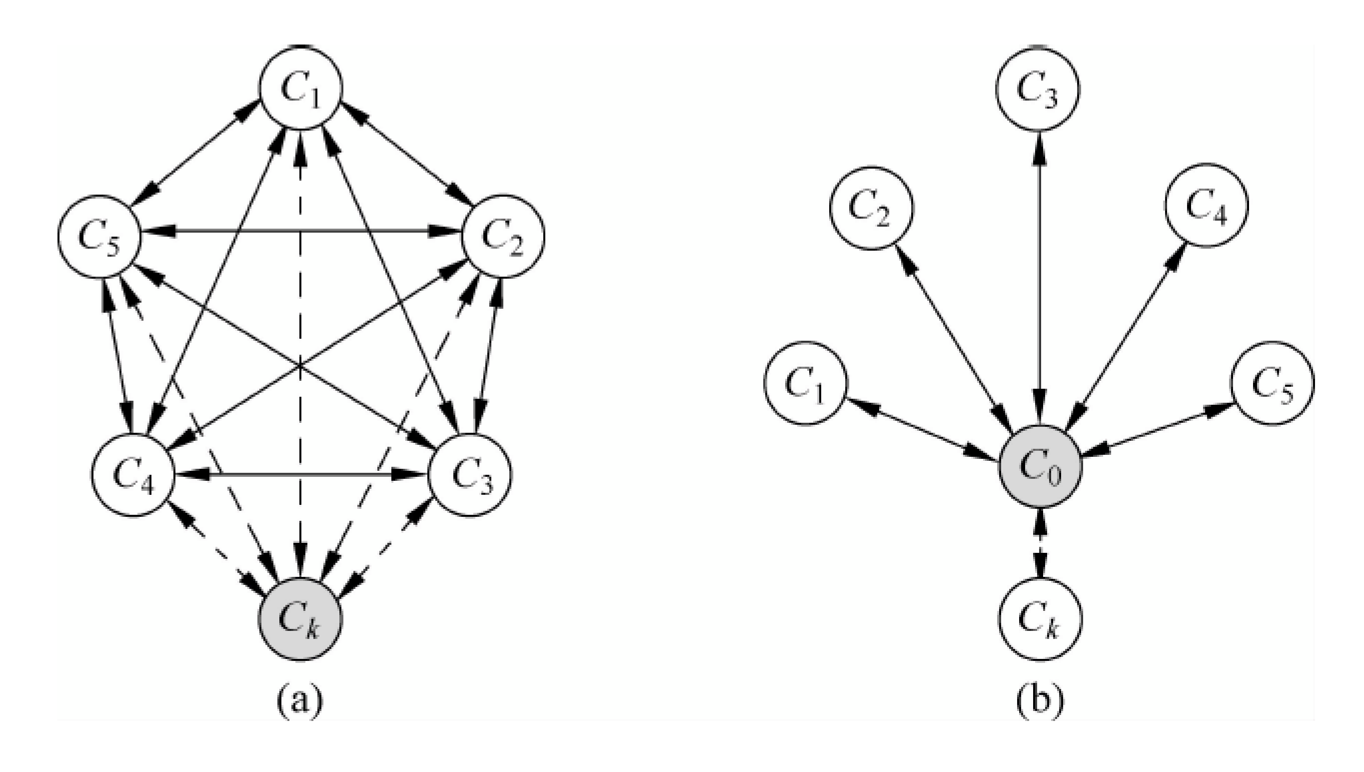


图 1.4　两种编码转换方案比较

表示层主要承担编码(数据表示法)转换的工作,例如,可采用 ASN.1 技术标准。

思考:举例说明计算机中需要相互转换的不同编码。

进一步思考:为什么 Internet 协议栈中没有表示层?

(7) 第七层为**应用层**(Application Layer)。应用层位于协议栈的最高层,顾名思义就是

支撑网络应用并与网络用户接口的层。应用层协议目的性很强，如用于下载文件、远程登录等。需要注意的是，应该把应用层与应用程序、用户界面等概念区分开来。

1.3　网络协议原理

根据 OSI 模型，网络协议就是对等层之间相互对话的规则。对话必须采用通信双方都能完全理解的"语言"，用相同的词汇和一致的语法、语义结构，即遵循同一种标准化的协议。

下面用一个例子说明协议是如何满足循序渐进式提出的数据传输需求的。

在本书的讨论中，经常会出现两个"著名人物"Alice 和 Bob，他们分别代表了通信双方 A 和 B，偶尔也会有 Cary 和 Henry 登场。在本例中，假定 Alice 需要发送数据给 Bob 和 Cary。

(1) Alice 发送一个消息给 Bob。

Alice 发送消息数据，Bob 接收。

(2) Alice 发送一个带标题的消息给 Bob。

标题和消息内容是两个不同的部分，Alice 应当让 Bob 收到后能够识别，因此，Alice 发送的数据报文被结构化为两个字段：标题字段和文本字段。可以设计专门的字符编码，分别用于指示标题和文本的起始位置，也可以用特定编码的分隔符来指示标题的结束和文本的开始。

(3) Alice 发送消息给 Bob，确保 Bob 正确接收。

Alice 和 Bob 首先约定如下协议：Alice 发送消息数据(可能包含标题)及其校验码；Bob 接收到数据后验证之，若正确，则回复响应报文，若错误，则回复拒绝报文；Alice 若收到 Bob 的响应报文，协议成功结束，若收到 Bob 的拒绝报文，则重新发送数据及其校验码。

然而，实际情况下还存在一种可能，即 Alice 长时间(永远)收不到 Bob 的响应，因为发送的数据或响应都可能丢失。为此，协议应加以改进：Alice 在发送数据后启动一个定时器(timer)，设定一个合理的超时时间(思考超时时间应如何确定)，当计数器回零且未收到响应报文，则重新发送数据、重启定时器。

但是到此为止，Alice 和 Bob 的协议仍然存在一个风险，假如 Bob 正确收到了数据，也发送了响应，只是响应报文丢失了，那么，Alice 的重发行为将造成 Bob 收到重复的数据，甚至可能多次重复。所以，协议还需要进一步完善：Alice 应给发送的数据打上序号，序号依次递增(可循环)；Bob 的响应也应带上对应的序号，当发现收到重复序号的数据时，只回复响应报文，数据作丢弃处理。发送序号的引入不仅避免了数据的重复接收，而且可以使协议能够支持大量数据的发送，即将数据分为较小的数据块，按序编号并依次发送，不易导致一个比特的传输错误而引起所有数据被丢弃。

(4) Alice 要可靠地发送一个大文件给 Bob。

既然之前已经设计了发送序号的协议机制，似乎大文件(即大量数据)的发送已经不成问题。但考虑信道质量较差的情况下，数据传输出错概率较高，例如 Alice 已经发送了序号为 0～6 的数据报文，但序号为 2 的报文出错了，Bob 回复拒绝 2 号数据报文，于是 Alice 就要重发序号为 2～6 的报文，通信效率十分低下。针对这一问题，协议可设计一种选择重发机制，即 Alice 只重发 Bob 拒绝的那个序号的报文，同时，Bob 保留正确接收的后续报文，

Bob 负责所有可能乱序到达的报文的有序拼接。

(5) Alice 要可靠地发送消息给 Bob 和 Cary。

显然,在上述协议的基础上,Alice 的发送数据中还需要增加地址字段,以指明接收者为 Bob 或 Cary;而 Bob 和 Cary 在响应报文中也要带有自己的地址信息,以便 Alice 能够区分是来自谁的响应。

上述过程还可以持续扩展下去,但已经可以让我们领会到通信协议设计的目的、基本原理和技术思路。例子中协议的功能包含了用于进行可靠传输的差错检测和恢复、用于实现信道复用的并发传输、用于提高通信效率的选择重发等。从中可以发现,协议中用于完成数据传输的成分只占很小比重,大部分工作量在于异常处理,运用"二八法则"来解释,就是协议用 80%以上的工作量来解决 20%以下的差错检测和恢复问题。付出如此高的代价是必要的,因为不处理这些异常情况,就不能达到 100%的可靠。由此可获得这样的启示:在高质量的通信信道上,应该使用尽可能简单而高效的协议(参见 6.2 节)。

1.3.1 协议数据单元

协议数据单元(Protocol Data Unit,PDU)是构成计算机网络协议的细胞,从这个意义上说,PDU 的数据结构定义就是协议的 DNA。事实上,ATM 协议的 PDU 即被称为 cell。然而,协议并非 PDU"细胞"的堆砌,PDU 的生命周期其实很短暂,始于发送、终于接收。

如图 1.5 所示,PDU 是一个结构化数据块,由控制头和承载数据两大部分组成。PDU 控制头为结构化数据,分为多个字段,可为各种字段长度和类型(数值、逻辑、枚举等),分别用于表示地址、序号、长度、流量控制、PDU 类型等协议要素;部分控制型 PDU 并没有承载数据部分,数据型 PDU 的承载数据部分为用户数据或上层 PDU,一般为非结构化数据,进行透明传输(搬运)。

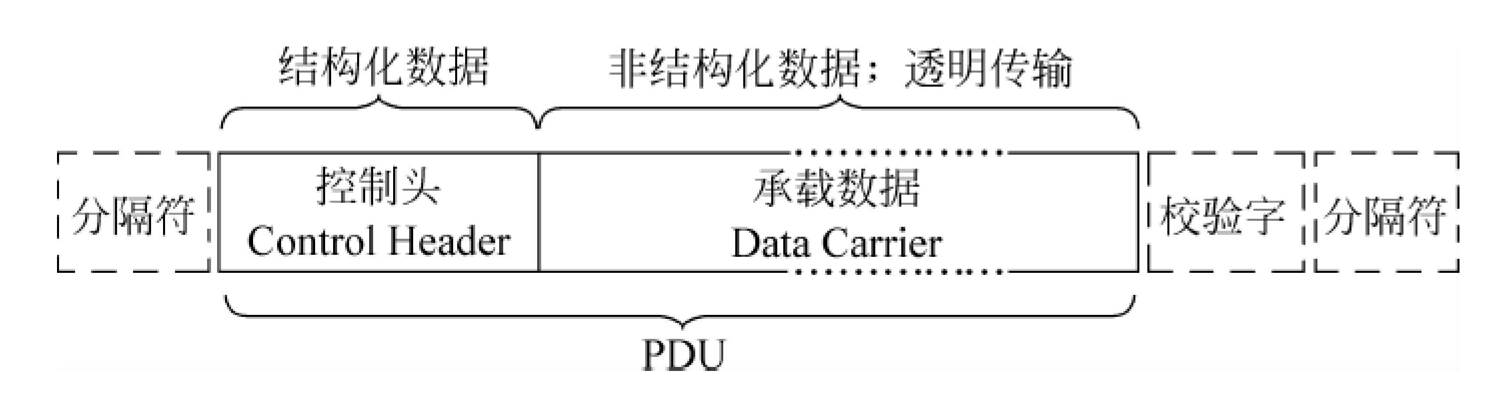


图 1.5 PDU 数据结构示意

图 1.6 展示了用户数据从应用层开始依次往下"穿透"每一层的情形。

每个协议层均为上层数据的发送构造 PDU,并把上层数据(上层的 PDU)作为整体来承载,而并不关心数据的构造和内容,这样,当接收端的对等协议层收到 PDU 后,剥离本层的控制头部分,即可原封不动地往上层递交数据。图 1.6 中的递归函数程序即说明了自上而下的 PDU 封装发送过程。

一般地,计算机网络协议 PDU 分为两种类型。

(1) 控制(Control)PDU——由本协议层生成,用于建立和拆除虚电路、流量控制、差错检测和恢复、状态报告等面向连接的管理业务。

(2) 数据(Data)PDU——承载上层协议的 PDU,实现有序的、透明的数据传输。

有些协议(尤其是面向非连接的协议)则把 PDU 类型进行了简化,仅采用一种 PDU,把

有限的控制和数据传输两种功能结合在一起，从而使协议变得非常简化，可以在很大程度上提高网络通信的效率（试比较简化 PDU 的思想与 RISC 计算机的技术思路）。

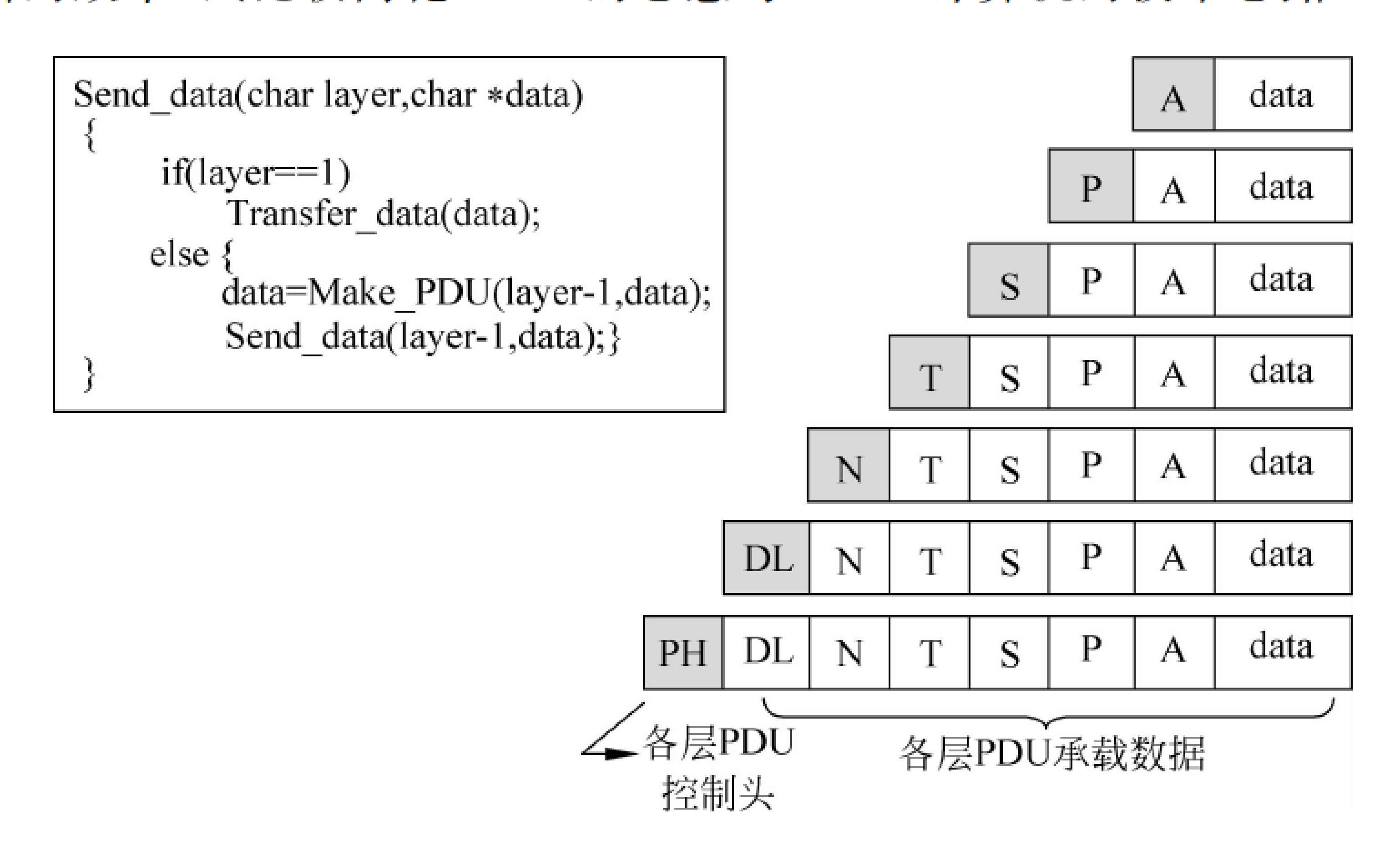


图 1.6　OSI 各层 PDU 控制头和用户数据

1.3.2　协议通信规程

通信规程阐述了两个通信实体（对等层）之间的对话方式，即一系列相互交换 PDU 的约定。这些约定包括：如何通过交换控制 PDU 来管理连接、检测错误与故障恢复；如何通过交换数据 PDU 来收发数据、分割和拼接数据；如何结合控制 PDU 来进行合理的流量控制等。

图 1.7 描述了一条虚电路的建立过程（步骤①～⑥）以及与相邻层之间的关系、与服务访问点、原语和 PDU 之间的关系。

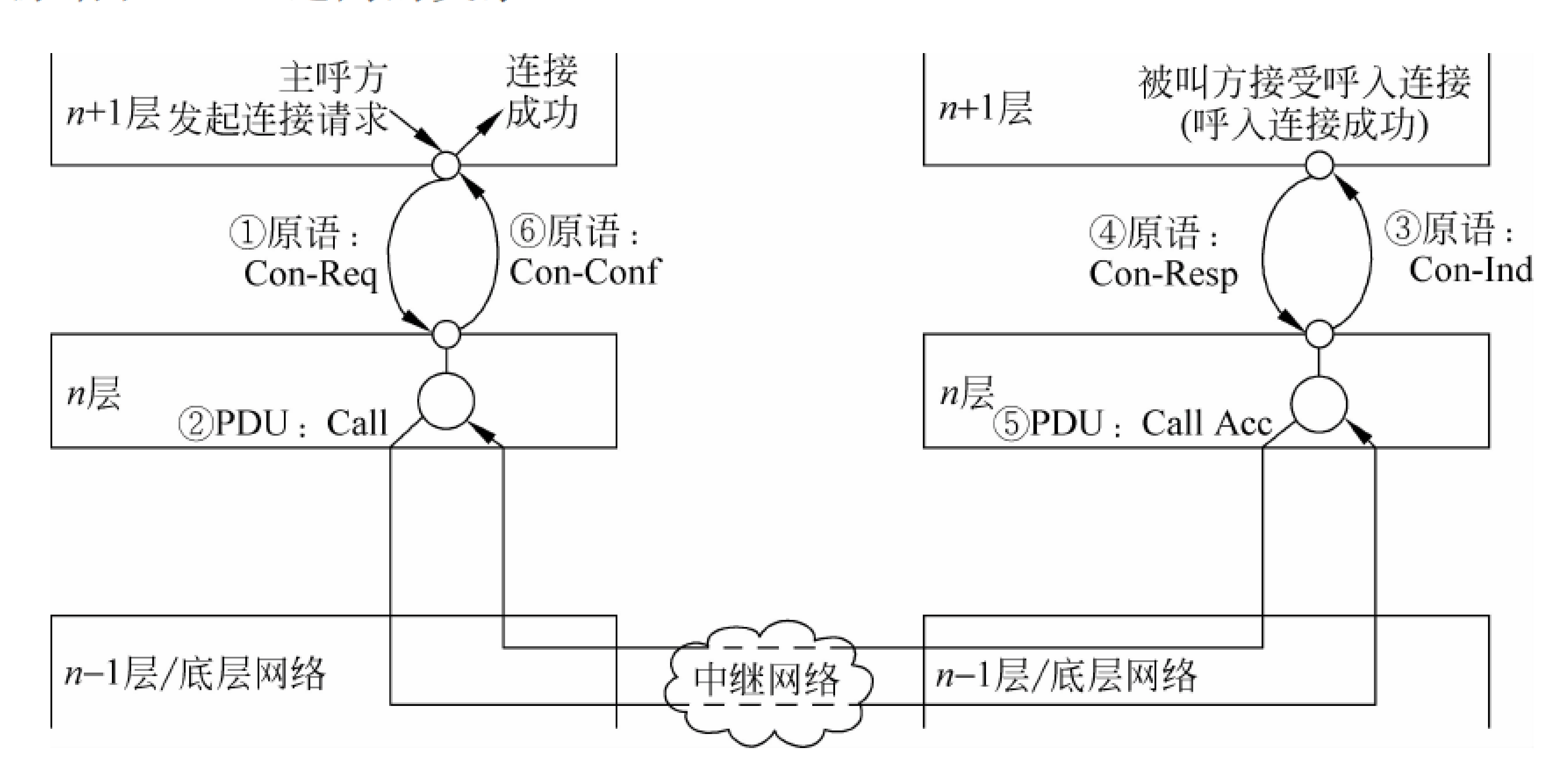


图 1.7　呼叫建立过程中的原语与 PDU

通信规程不仅在数据链路层以上的协议层次中使用，物理层的数据传输控制同样需要通信规程的支持。

图 1.8 展示了计算机中常用的 EIA RS-232 接口（通常称为 com 或 console 接口，与 ITU-T V.24 和 V.28 相似）的引脚定义，DTE 方为计算机终端，DCE 方为网络接入设备（如 Modem）。

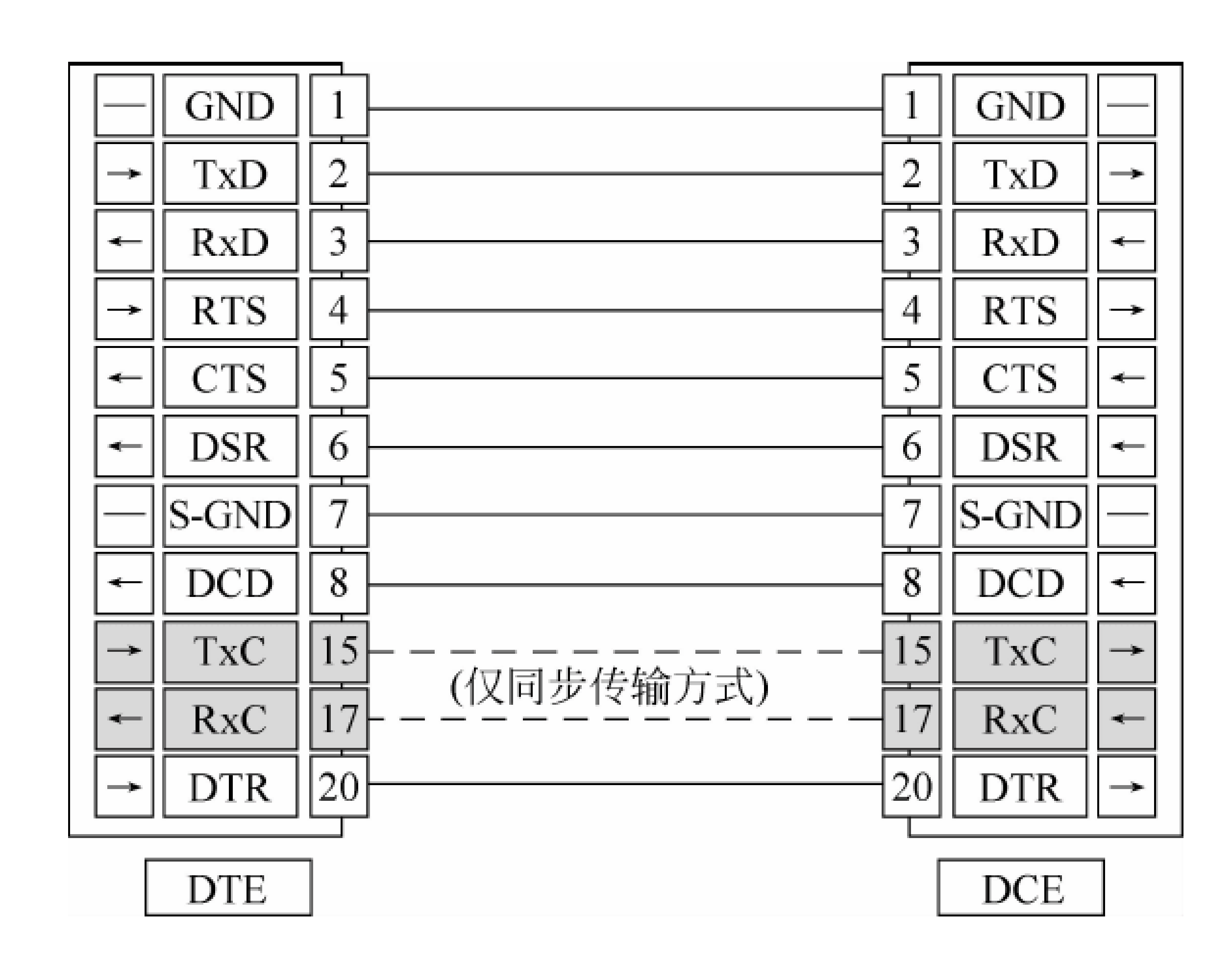


图 1.8　RS-232 接口引脚定义

RS-232 接口各个引脚的信号之间的关系如图 1.9 所示，通信双方通过信号电平的改变来实现各种控制功能和数据传输功能。

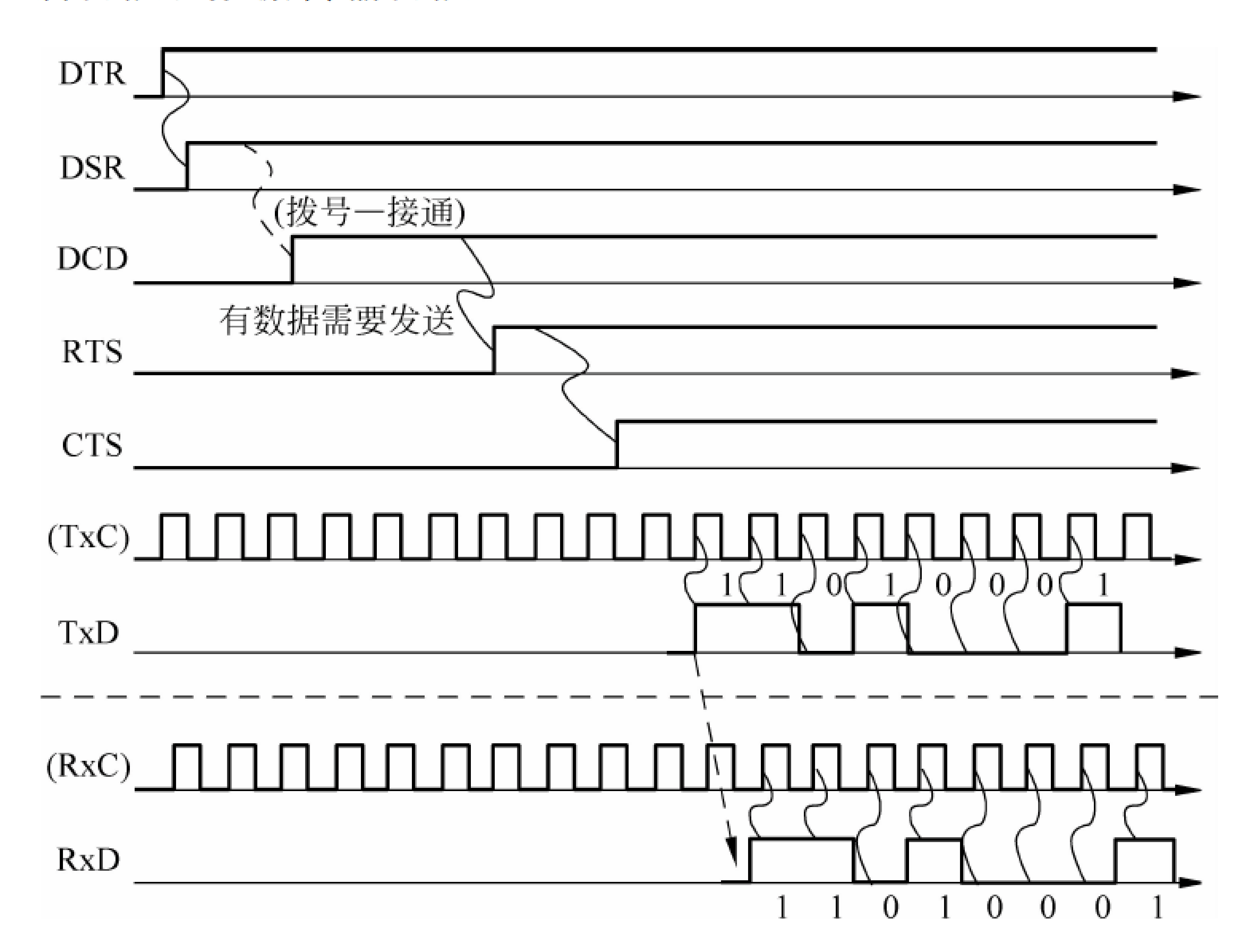


图 1.9　异步串行通信信号关系示意

RS-232 接口的通信规程简要说明如下。

(1) 当 DTE 要求建立连接时，首先给出 DTR 信号，DCE 通过 DSR 信号进行响应；DCE 尝试与通信目的方建立物理连接(例如采用拨号方式)，一旦成功，即以 DCD 信号告知 DTE。

(2) 当 DTE 需要发送数据时，应升起 RTS 信号；若 DCE 同意接收数据，则以 CTS 高电平表示接受，如果 DCE 不同意(或者未准备好)接收数据，则保持 CTS 低电平，DTE 就不能发送数据，否则升起 CTS 信号准备接收数据。以此规程可以实现物理层的流量控制。

(3) 当 DTE 完成数据发送后，可取消 DTR 信号，DCE 了解其意图后释放物理连接。

思考：如何采用 RS-232 接口“背靠背”连接两个 DTE(终端)设备？如果不考虑流量控制，一个最简化的 RS-232 接口是什么样的？

1.3.3　网络协议类型

按照数据比特的传输方式，可分为**并行传输**(Parallel Communication)和**串行传输**(Serial Communication)两种通信接口。并行传输指同时发送一个 8b(或更多比特)的数据，如打印机接口、计算机内部数据总线等。但网络通信需要的距离较远，并行传输需要较多的数据信号线(或信道)，对通信资源的要求过高，所以，不论是有线信道还是无线信道，计算机网络总是采用串行传输方式，即按比特依次传输。

串行通信进一步可分为**异步串行通信**(Asynchronous Serial Communication)和**同步串行通信**(Synchronous Serial Communication)两种类型。两者的差异主要在于通信双方是否采用完全相同的时钟(clock)。如图 1.9 所示，在异步串行通信过程中，没有专门的时钟线路(同样可以节省线路资源)，数据发送方根据发送时钟(TxC)来发送串行的比特流，每个时钟周期发送一个比特，接收方则根据一致的时钟频率(接收时钟 RxC)来提取比特。然而，由于发送方和接收方所用的时钟都是由本身计算机生成，因此 TxC 和 RxC 必然会有差异：时钟相位上的差异可以通过接收方使用合理的采样技术来调整；但是，时钟频率的差异，即便很微小，积累效应也会很大，严重到一定程度无疑会影响到接收方的比特采样的准确性。为此，异步串行通信采用两个办法来应对时钟误差。

(1) 每个字节以起始比特开始，停止比特结束，通过起始和停止比特的调整，把误差积累限制在 5～8 比特字节范围内。

(2) 保证在一个时钟周期范围内都能正确采样比特，从而在一个字节范围内允许一定的误差存在。

设 TxC 和 RxC 的时钟周期分别为 T_s 和 T_r。发送 8b 数据，加上起始和停止比特，共需发送 10b。

不失一般性，设 $T_s < T_r$，(即接收时钟稍慢)，需满足

$$9 \times T_r < 10 \times T_s$$

解得

$$\frac{T_r - T_s}{T_s} < \frac{1}{9} \approx 11.1\%$$

说明 TxC 和 RxC 允许有约一成的误差。

在网络应用系统中，通信双方或多方之间的事务处理、协议流程等也经常会用到同步和异步的概念，但不一定是基于时钟的，往往是基于状态的。假如在某个事件的触发下，双方进入一致的状态，并等待进一步发生的事件(如接收期望的数据)，则属于同步控制方式；假如不依赖于状态，不作停等，而是继续执行其他事务，一旦事件发生仍然可以进行对应的处理，则属于异步控制方式。两种方式没有绝对的优劣之分，应根据需要来分别运用。

除了时钟同步上的差别，异步串行方式和同步串行方式分别提供不同的数据传输模式。异步串行传输适用于面向字符流的数据传输，以字符为单位依次发送数据，字符间可以“停顿”(即无任何数据)，主要用于字符串(如命令串)或**数据报**(datagram)的传输，对应于**字符型协议**和**报文交换**(Datagram Switching)方式；同步串行传输则用于面向比特流的数据传

输,即提供面向分组(packet)的传输,对应于**分组型协议和分组交换**(Packet Switching)方式。

比特流传输的基本原理是:信道上比特紧密排列传输,分组内部没有空隙或停顿;采用特定的表示起始或结束标志符(flag)的比特模式(如0111,1110),用以分隔分组;通过发送方零插入、接收方零删除的方法,避免标志符的影响,实现数据的透明传输(详见1.4节)。

当需要在异步通信线路(或接口)上传输分组型数据时,可采用BSC、SLIP、PPP等协议进行转换,通过转义字符的插入和删除来保证数据的透明传输(详见1.5节)。

思考:考察传统电话网拨号上网所使用的调制解调器(Modem)采用的AT命令集及其操作规程。

1.4 BSC和SLIP

二进制同步通信(Binary Synchronous Communication,BSC)协议是经典的面向字符的同步协议,由IBM公司提出,是最早的同步协议。

值得注意的是,BSC协议是在异步串行通信线路上传输的,但命名为同步,说明BSC协议可用于传输控制信息和数据分组。

如表1.1所示,BSC协议通过定义一系列控制字符的方法来实现控制命令、响应的传输,结构化数据传输和透明数据传输。

表1.1 BSC协议控制字符

控制字符	SOH	STX	ETX	EOT	ENQ	ACK	DEL	NAK	SYN	ETB
名称与含义	序始	文始	文终	送毕	询问	确认	转义	否认	同步	块终
ASCII码值	0x01	0x02	0x03	0x04	0x05	0x06	0x10	0x15	0x16	0x17
EBCDIC码值	0x01	0x02	0x03	0x37	0x2d	0x2e	0x10	0x3d	0x32	0x26

BSC协议的报文一般由报头和文本组成。当信道上无数据发送时,可以用同步字符(SYN)填充,同步字符同时起到分隔不同报文的作用。两个报文间至少需要两个SYN,使双方实现同步。报头以SOH开始,文本由STX开始,两者均以其他控制字符结束。报头中可包含识别符、地址等信息。

BSC协议有数据和控制两类报文,控制报文又可分为正向控制和反向控制两种,每一种报文中至少包含一个控制字符。例如,ENQ用于轮询功能,在多站结构中,被轮询的站点地址位于ENQ字符前;EOT用于标志报文交换的结束,在两站点间拆除逻辑链路。

所有数据报文在块终符(ETB)或文终符(ETX)之后附加块校验符(Block Check Character,BCC),BCC可采用垂直奇偶校验或16b的CRC码,校验范围从STX开始到ETX或ETB为止。长度不超过限制的文本数据可只用一个数据报文发送;较长的文本数据则可分作多块,用多个数据报文发送。接收方对于每一个收到的数据报文都要给予确认。当发送方收到返回的肯定确认(ACK)后,才能发送下一个数据报文;而如果收到否定确认(NAK),则说明报文有误而被拒绝,应重新发送。

一个典型的BSC协议报文结构如下:

…[SYN][SYN][SOH]…报头…[STX]…文本…[ETX][BCC][SYN]…

显然，当文本数据中包含与控制字符相同编码的内容时，就会出现失步，造成协议处理的混乱。为了实现数据的透明传输，需要通过转义字符(DLE)来解决编码二义性问题。方法是：除了同步字符外，其他控制字符(设为[xxx])在发送时均采用[DLE]-[xxx]的表达方式；当文本中出现 DLE 编码(实际为数据)时，则采用[DLE]-[DLE]进行发送。

接收方的处理相当简单：当收到单个的 DLE 字符时，其后必为控制字符；当收到一对 DLE 字符时，去掉一个，另一个为数据字符。

思考：*为什么不需要处理(转义)文本数据中的 SYN 字符编码？为什么不采用统一在文本数据中每遇到控制字符就插入一个 DLE 字符的方法？*

串行线路互联网协议(Serial Line Internet Protocol，SLIP)是 Internet 的专用协议(RFC 1055)，用于在串行通信线路(如低速拨号网络)上传输分组型的 IP 报文。

SLIP 与 BSC 协议采用类似的工作原理，但由于其目的很单纯，就是封装 IP 报文，起到转换作用，因此不需要复杂的控制字符，只需要实现报文间的分隔和数据的透明传输。为此，SLIP 协议定义 END(0xc0)为报文开始和结束符，ESC(0xdb)为转义字符。当 IP 报文中出现 END 编码时，传输 0xdb-0xdc；当 IP 报文中出现 ESC 编码时，传输 0xdb-0xdd。

1.5 LAP 协议

链路接入规程(Link Access Procedure，LAP)代表了一系列的数据链路层协议，由**同步数据链路控制**(Synchronous Data Link Control，SDLC)和**高级数据链路控制**(High-level Data Link Control，HDLC)两种协议发展而来，之后又衍生出用于 X.25 分组交换网络(PSDN)的 LAPB(LAP Balanced for X.25)协议、用于 ISDN 的 LAPD(LAP on the D channel)协议和用于 FR 网络的 LAPF(LAP for Frame-mode Services)协议。

LAP 是面向比特流的分组型协议，在同步串行信道上以连续的比特传输每一个分组，分组间用标志 F(Flag)作为开头和结尾(开头、结尾可以共用一个 F)。标志 F 编码为 0111，1110(即 0x7e)，应理解为一个“2 个 0 之间连续 6 个 1 的比特流模式”。为了达到数据透明传输的目的，一旦分组比特流中出现 F 编码，就需要设法打破这一模式，防止接收方因误解而进行错误操作。采取的方法是零比特插入法：分组比特流中每出现连续的 5 个 1，就自动插入一个 0；接收方自动删除连续 5 个 1 后面的 0。例如，有数据帧{0x73，0xf5，0xf6，0x7e，0xfa，0x3e}，发送方总共应插入 5 个 0，而不是表面上看的一个 0(思考为什么)。

进一步思考：*当接收方收到信道上传来的以下比特串后应如何处理？*

…0111100111110111001111110011111000101111100110110000…

LAP/HDLC 协议有非平衡配置和平衡配置两种模式。

非平衡配置中由主站(Primary Station)控制整个链路的工作，而次站或从站(Secondary Station)不能主动发送信息，只有当主站轮询或探询(polling)时才可以回复。主站发送的帧称为命令帧(command)，从站发出的帧称为响应帧(response)。在一个特定的通信系统中，主站是唯一的，从站可以有多个。

非平衡 LAP/HDLC 协议的半双工通信方式较为适合总线型共享信道，可避免发送冲突的发生，同时可应用于卫星和各个地面站之间的通信。

平衡配置则采用复合站(Combined Station)方式，同时具有主站和从站的功能，通信双

方对称部署,相当于将两条半双工信道合成为一条全双工信道。但衍生出的 LAPB 等协议已经直接采用对等的通信方式,没有主站和从站的区分,双方都可以主动发送命令和响应,成为彻底的全双工方式。

LAP 系列协议的帧结构有很大相似性,主要差别在地址(address)和控制(control)字段,帧类型也不尽相同。例如 LAPD 协议将两个字段合为一个进行编码,但工作原理和主要功能比较一致,主要用以解决在物理信道上进行可靠数据传输的问题。

以 HDLC 协议为例,帧结构如图 1.10 所示。

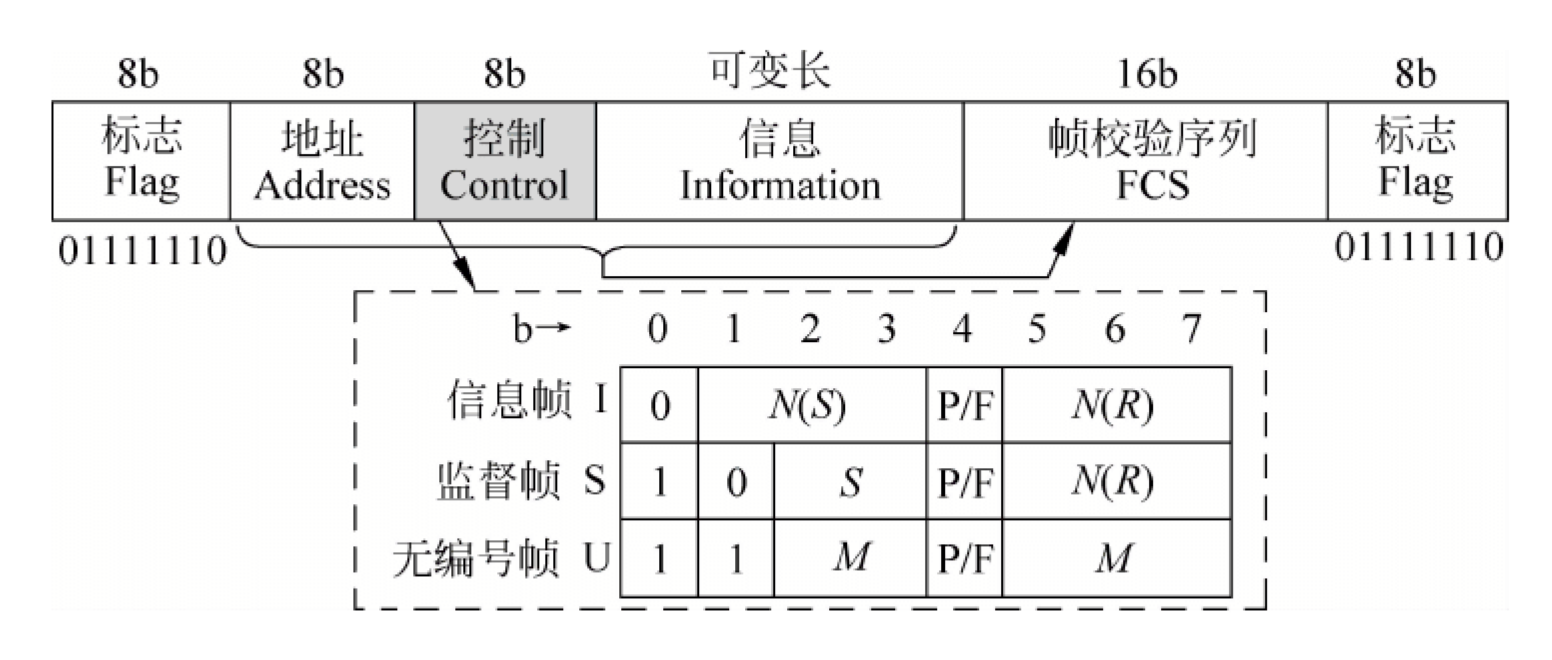


图 1.10 HDLC 帧结构

地址字段用以标识一个站点,通过事先的约定来确定地址码。在非平衡模式下,地址字段总是填入次站地址;而在平衡模式下,地址字段总是填入确认站的地址(即命令帧中填对方地址,响应/确认帧中填本方地址)。如果需要扩展地址字段,就以第 0 比特为地址扩展标志(EA),当 EA=0 时,表示后一个字节仍然为地址信息,而最后一个地址字节的扩展标志 EA=1。

控制字段用以区分不同的帧类型并携带协议控制信息。其中,P/F 位为探询/终止(polling/final)标志,主站发送的命令帧中 $P=1$ 表示要求对方立即响应,而对方发送的确认帧中 $F=1$ 表示数据发送完毕;$N(S)$和 $N(R)$分别为发送和接收序号,在 nb 编码方式下,序号空间为 $0\sim2^n-1$,进行循环编码;$N(R)$的含义是确认已经接收到($N(R)-1$) mod 2^n 和之前的帧、准备好接收编号为 $N(R)$的帧;信息帧中的 $N(R)$具有捎带确认(piggy-backing)的作用;S 和 M 字段是监督帧和无编号帧的类型码,如 $S(bit_3-bit_2)=00$ 为接收就绪(Receive Ready,RR),01 为接收未就绪(Receive Not Ready,RNR),10 为拒绝(Reject,REJ),11 为选择拒绝(Selective Reject,SREJ)。无编号帧有 SAM/SABM、UA、DISC 等,用于链路逻辑连接的建立和其他控制操作。

HDLC 采用多种技术手段来保障无差错数据传输,包括帧校验(差错检测)、帧确认和重发(差错恢复)等机制,还可用滑动窗口机制进行流量控制。

1.5.1 帧校验机制

帧校验序列(Frame Check Sequence,FCS)为 16b 数值,采用循环冗余码(Cyclic Redundancy Check,CRC)检错技术。CRC 码是一种线性分组码,编码和解码方法简单,检错和纠错能力强,能够达到 0.0047%以下漏检率,可检测出所有奇数个随机错误、长度小于等于生成多项式阶数的突发错误。

设传输的比特流为 M,需生成 nb 的 CRC 码 $R(x)$,则应采用 n 阶的生成多项式 $P(x)$。

例如，生成多项式 $P(x)=x^7+x^5+x+1$，对应二进制数即为 10100011。

CRC 码计算方式如下：

$$\frac{M\times 2^n}{P(x)}=Q\cdots R(x)$$

余数 $R(x)$就是 CRC 码（高位用二进制 0 补足 nb）。将该 CRC 码填入 FCS 字段进行传输，相当于传输 $S=M\times 2^n+R(x)$。

在串行传输链路上，当数据 M 按比特依次发送时，即可一边发送，一边进行 CRC 除法运算。这一过程可以如下表述（如图 1.11 所示）。

（1）采用一个 $n+1$ 比特的发送寄存器和一个 $n+1$ 比特的除法寄存器，均进行逻辑左移操作；在待发送数据的尾部添加 n 个二进制 0。

（2）将一个待发送数据的比特同时移入发送寄存器和除法寄存器的最低位（添加的 n 个 0 将只移入除法寄存器）；从发送寄存器左移溢出的最高位比特可进行发送。

（3）当除法寄存器满时，用 $P(x)$作为除数进行一次按位除法运算（按位异或），得到的中间值放入除法寄存器，并左移去掉高位连续的 0。

（4）返回第（2）步，直到添加的所有 n 个 0 都已参与运算。

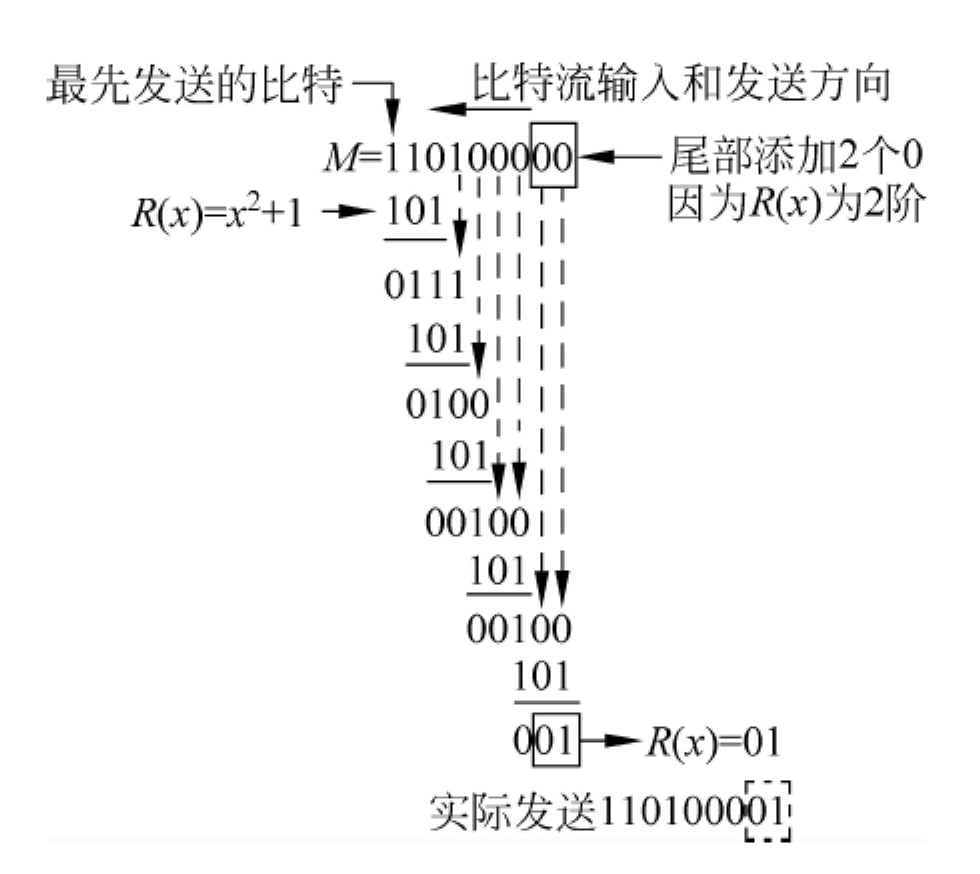


图 1.11　CRC 按位除法示例

（5）取除法寄存器的低 n 比特（即余数 $R(x)$），从高位起依次移入发送寄存器最低位，左移发送，直到发送寄存器为空。发送完毕。

接收方采用同样的生成多项式 $P(x)$进行按位除法运算：

$$\frac{S}{P(x)}=\frac{M\times 2^n+R(x)}{P(x)}=q\cdots r$$

若余数 $r=0$，表示接收无误，否则表示有误码。在某些特殊情况下，传输误码可能也会碰巧得到 $r=0$，但只要 $P(x)$选择得当，可以把漏检率降到极小。较常用的 $P(x)$多项式有

$$\text{CRC-16}=x^{16}+x^{15}+x^2+1$$
$$\text{CRC-CCITT}=x^{16}+x^{12}+x^5+1$$
$$\text{CRC-32}=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$$

思考：*数据发送时，应该先计算 CRC 码还是先进行零比特插入？*

1.5.2　帧确认和重发机制

协议对发送的信息帧进行 $N(S)$编号，接收方如果正确接收到一个帧，就以 $N(R)=(N(S)+1)\bmod 2^n$进行确认，如果一个帧接收错误或根本没有到达，接收方不作确认，那么发送方在一定时间后（确认计时器超时）就要考虑重发。

在某些情况下，接收方可以发现帧丢失的情况。例如，接收方设置期望收到下一帧编号的寄存器 $V(R)$，若序号缺失，即接收到的 $N(S)>V(R)(\bmod 2^n)$，就可以及时报告给发送

方,不必等待确认计时器超时才能发现错误。假如发现收到的帧序号超出了规定范围(如窗口),应作为严重故障处理(如进行复位)。

在停等式(或称乒乓式)的**自动重发请求**(Automatic Repeat reQuest,ARQ)协议中,每个信息帧都要进行确认;在连续ARQ协议中,发送方不必等待确认,可以连续发送多个信息帧,接收方也可以一次性确认多个帧,可以提高协议工作效率,提高信道带宽利用率。

如果接收方发现传输差错,可以通过发送REJ帧请求重发。发送方确认计时器超时后,将认定帧已经丢失,主动进行重发。简单重发机制规定从出错的帧开始,后面已经发送过的帧都要按顺序重发一遍,而不管这些帧是否已经发送成功。

对于一些带宽资源比较宝贵的链路,简单重发会产生较大浪费,带来的延迟也可能较大,可采用选择重发机制,即用SREJ帧指定需要重发的帧,发送方只重发指定帧,其他已发送的帧不受影响。在这种情况下,接收方需要缓存差错帧后收到的所有正确帧,会产生系统复杂性增加、存储空间消耗大等问题。

1.5.3 滑动窗口机制

滑动窗口(Sliding Window)是指在发送端设定一个发送序号区间,在这个区间内,可以不经确认连续发送信息。每发送一个信息帧,窗口值减1,当窗口值为0时,发送暂时停止。当发送端收到接收端发来的确认后,将确认的帧数加入窗口值,窗口同时向前滑动,可以继续发送信息帧(如图1.12所示)。

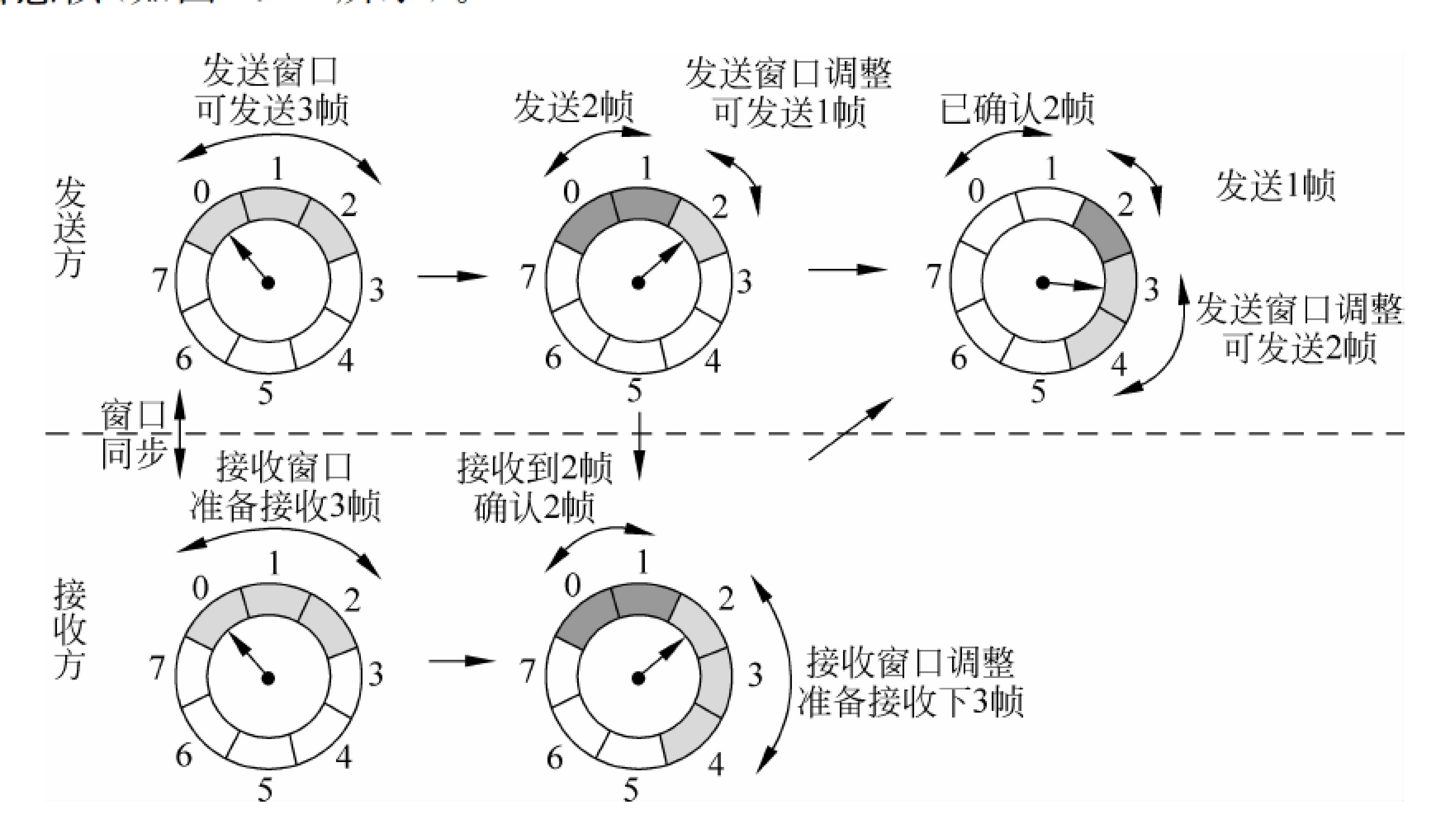


图1.12　窗口流控示意

显然,窗口机制可以起到端到端流量控制的作用。最大窗口值 K_{max} 由通信双方事先约定,满足接收端的接收能力,同时考虑到通信线路的延迟等因素,使信道可以被充分利用。接收方也可以设定接收窗口,一方面据此检查帧序号是否越界,另一方面可以进行批量确认。

设帧序号为 nb,编码空间为 $0\sim2^n-1$,那么必有

$$0 < K_{max} < 2^n$$

$K_{max}>0$ 很显然,$K_{max}<2^n$ 可采用反证法证明。

先考虑 $K_{max}=2^n$ 的情况，假设发送端用满窗口发送了 2^n 个帧。第一种情况，如果这 2^n 个帧都丢失了，那么来自接收端的 $N(R)$（在监督帧或信息帧中携带）会被发送端误以为是对这些已丢失帧的确认；第二种情况，如果这 2^n 个帧正确到达，但所有确认帧（可能只有一个批量确认帧）丢失，发送方就会超时重发，而接收端并不知道是重复发送的，从而引发错误。

再考虑 $K_{max}>2^n$ 的情况，假设发送端发送了 2^n 个以上的帧，由于发送序号是循环使用的，则必有两个信息帧的 $N(S)$ 相同，那么一个确认的 $N(R)$ 就可能会出现二义性。

在序号编码空间的规划方面，设从发送端到接收端的发送时延为 T 秒（例如卫星地面站间通过卫星转发数据的单跳时延约为 270ms），双向传输速率均为 Cb/s，信息帧长度为 LB，接收端收到第一个信息帧后立即进行确认，假定确认帧长度、计算机处理时间等忽略不计，那么，为了充分利用信道资源，发送窗口 K 要足够大，使发送端可以连续发送信息帧而无须等待确认，相当于要让信息帧的比特填满来回信道（注意不要遗漏发送第一个信息帧需要的数据传输时间），因此有以下关系式：

$$K=\frac{\left(2T+\frac{8L}{C}\right)\times C}{8L}=\frac{TC}{4L}+1$$

得到 K 值后，再根据以上讨论的 K_{max} 与帧序号编码空间的约束要求，即可以确定帧序号的编码比特数。例如，如果 $K=4$，序号应为 3b 编码；如果 $K=67$，序号应为 7b 编码。

CHAPTER 2

第2章 Ethernet协议

以太网(Ethernet)是最常见、最常用的网络之一。以太网以性能优异、价格低廉、使用便捷著称。在校园、机关、企业、家庭以及城市的各个角落几乎都部署了以太网,我们使用的各种计算机设备、移动通信终端都有以太网接口,便于随时随地接入网络。

学习掌握以太网技术原理,应先从分析共享网络开始。

2.1 共享网络原理

共享网络(shared network)是指所有网络设备都使用同一个通信信道。

以课堂为例,所有学生和教师在同一个教室空间中,通过周围的空气传播声音。每个人都是通信实体,声音就是传输的数据,而空气就是通信媒介,形成类似计算机网络的共享网络系统。考察这个系统,可以观察到存在以下两个现象。

(1) 教室里任何一个人讲话,其他人都可以听到。即使是点名提问,别人也在听,只不过未被点到的人不响应罢了。

(2) 如果有两个人同时讲话,就有可能谁也听不清楚。

我们不能因此而得出这个"系统有问题"的冒失结论,只能说这个系统具有自身的特点,使用者应该想办法去适应它。所以,为了维持系统的正常运行,即保持良好的课堂秩序,这个系统需要一定的规则。例如,如下两种规则执行任何一条都是有效的。

(1) 规则一:由教师主导讲话,除非被点名发言,否则学生不能讲话。

(2) 规则二:教师和学生都可以自主发言,参与讨论,但每个人都应保持基本的礼节,别人讲话时不能插嘴,发生异口同声的情况时可相互谦让或协商。

课堂这个系统还有一个良好的特点,就是每个个体都是独立、平等的,晚到的学生可以悄悄入座,早退的学生可以偷偷离席,不会影响系统的正常运行。

共享网络具有类似的特点，当然计算机没有礼节的概念，于是需要为联网的计算机设备设计满足共享网络需求的算法，以避免发送数据的冲突，以及一旦发生冲突也能尽快发现和恢复。

2.1.1　时钟同步方案

假定共享网络上所有计算机设备 $S_1S_2\cdots S_n$ 有完全一致的时钟，即时钟同步，则采用如图 2.1 所示的方法，先将时间轴等分为一个个的**时间片**(Time Slice)，设为 T_δ，并规定每个计算机设备每次可使用一个时间片，然后把发送权交给下一个计算机设备，依次类推。n 个时间片为一个周期，循环往复。其实，这种方式就是**时分复用**(Time Division Multiplexing, TDM)技术。

图 2.1　时分复用的共享网络方案示意

时钟同步方案具有如下优点。

(1) 控制方法十分直观而简单，不需要执行专门的协议；

(2) 所有设备都是平等的，无须控制中心这样一种特殊设备；

(3) 任何一台设备损坏(除非是干扰信道的故障)基本不会影响网络的正常运行。

时钟同步方案具有如下缺点。

(1) 通信资源分配看似完全“公平”，实质上效率低下，因为不论设备是否需要发送数据，时间一到，就拥有发送权，信道利用率很低。在极端情况下，发送任务可能集中在一台设备上，则信道利用率仅为 $1/n$。而且为避免一个完整的数据报文发送完成前时间片到时(成为碎片报文)，时间片长度会设置较大余量，结果大部分时间片往往使用不足，效率进一步降低。

(2) 更糟的情况是，第 $n+1$ 个设备的加入是个艰巨的任务，减去其中一个设备的难度也不小，需要其他设备做出相应的调整，会影响到系统中所有的设备。

(3) 最糟的情况是，方案设计是在假定所有的设备都能达到完全时钟同步的前提下，然而实现同步非常困难，一般需要同一时钟源，需要对时钟信号传输距离进行补偿，需要使用昂贵的对时设备(如卫星时钟)或建设专门的时钟同步网络。

可见，这一方案的缺点所造成的困难远远大于其不太明显的优点所带来的好处，因此仅有理论上探讨的价值，实际应用的可行性很小。

2.1.2　异步轮流方案

既然时钟同步方案采用的均分时间片方式存在严重浪费资源等缺陷，不妨换一个思路，采用异步控制方法，使宝贵的通信资源不是事先被均分，而是体现按需分配思想，可能更具有合理性。

如图 2.2 所示，计算机设备 $S_1S_2\cdots S_n$ 依次轮流发送数据，约定从 S_1 开始，每个设备每次发送一个数据报文，然后交由下一个设备发送，依次轮流下去，并循环往复。

有一个显而易见的问题：某台计算机设备被轮到时，却没有数据可发，应如何处理？可能的办法是：不做任何动作，或发送一个空报文，或稍作等待。

试想：“下一台”计算机设备是如何知道“上一台”计算机设备已经发送完毕了？提示：可以从数据报文构成的角度去考虑。

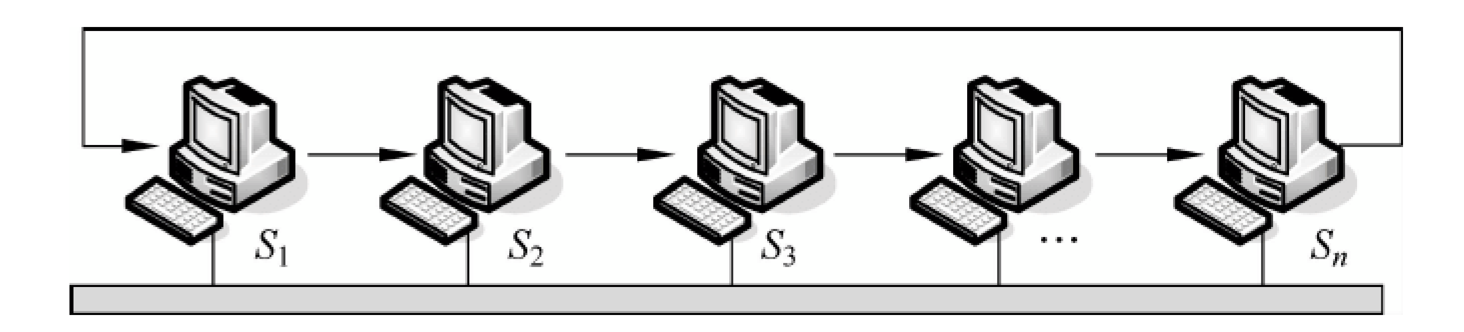


图 2.2　异步轮询的共享网络方案示意

异步轮流方案具有如下优点。

(1) 控制方法十分直观而简单,不需要执行专门的协议。

(2) 所有设备都是平等的,无须控制中心这样一种特殊设备。

(3) 任何一台设备损坏(除非是干扰信道的故障)基本不会影响网络的正常运行。

(4) 信道空闲等待时间较少、利用效率较高,遵循了信道资源“按需分配”原则。

异步轮流方案具有如下缺点。

(1) 严格的轮流发送规则还是容易造成不必要的信道资源浪费,尤其是在发送任务不均衡的情形下,信道利用率将有较大程度的降低。

(2) 计算机设备的增减仍然会对系统中其他设备产生影响,可能需要修改所有其他设备的配置。

异步轮流方案的最大缺陷在于把问题过于简单化,它试图通过不同的技术路线解决时钟同步方案存在的问题,实际上并没有完全解决,还带来了难以控制和管理的新问题。因此,异步轮流方案也仅限于理论探讨上的意义。

2.1.3　主从轮询方案

从所有计算机设备中选取(指定)一个设备 H 为主设备,或称主站(Primary Station),其他设备 $S_1S_2\cdots S_n$均为从设备(Slave)或称从站(Secondary Station),如图 2.3 所示,可据此设计共享网络的主从轮询方案。

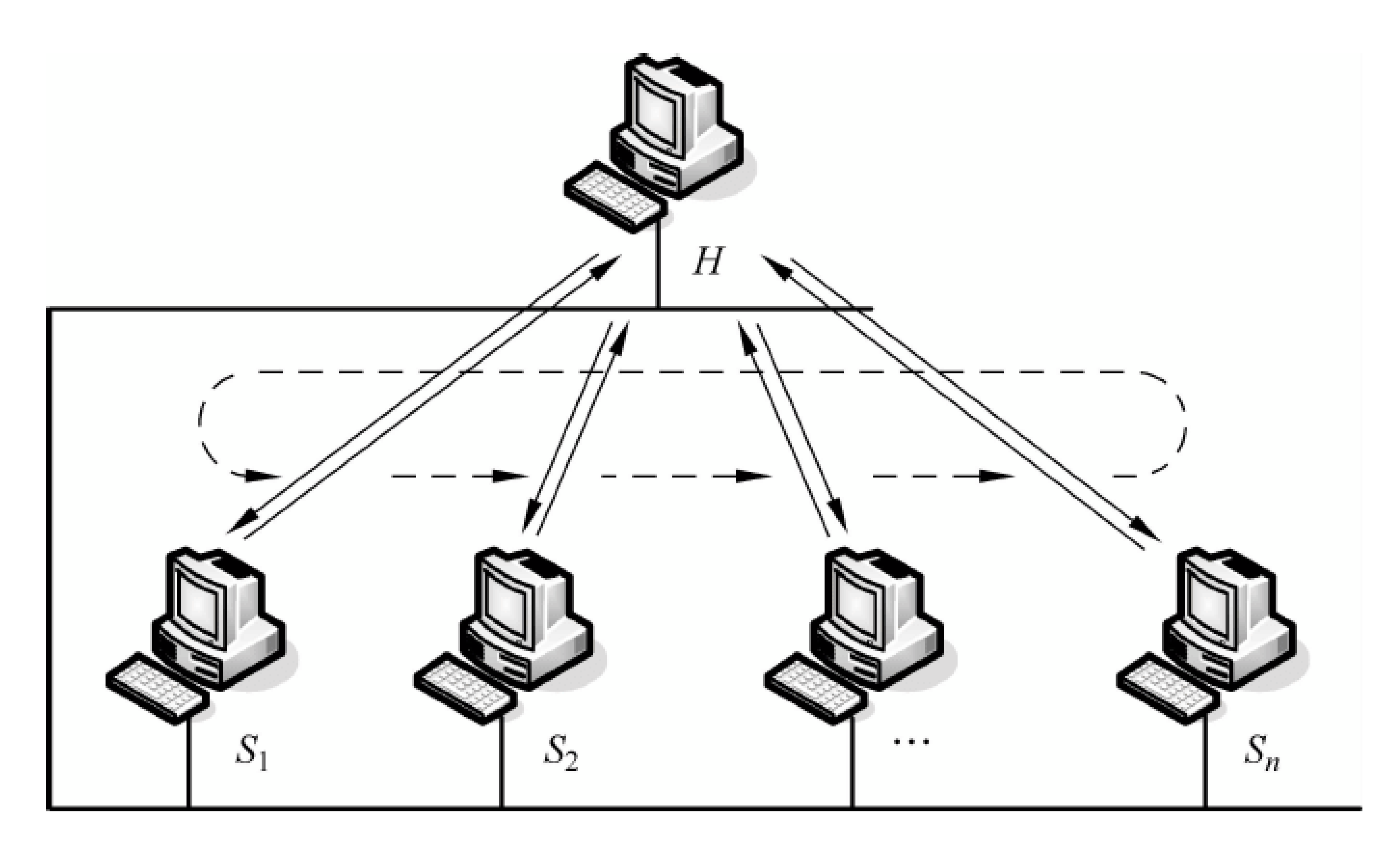


图 2.3　主从轮询的共享网络方案示意

主站 H 可向所有从站发送数据,从站通常只能向主站发送数据,这个过程是这样完成的:主站向从站发送探询报文,询问从站是否有数据发送,若有,从站发送数据,否则发送空报文或其他控制报文。

思考：

(1) 按照上述算法，从站间如何实现相互通信？

(2) 为提高通信效率，从站被允许发送数据后，能否连续发送多个数据帧？如何进行？

(3) 能否改进协议使得从站间可以直接通信？需要满足什么条件？

考察卫星通信网络(Satellite Network)，卫星是天然的主站，地面站均为从站，因为地面站的碟形天线正对卫星，只能与卫星通信，而地面站之间相隔遥远，难以相互通信。再例如安保网络中，星罗棋布的安保探头(包括摄像机、各式传感器)需要实时向监控中心报告探测到的状况，监控主机自然是主站，安保监测点部署的微型控制器就是从站。

选择主站在技术上通常基于性能需求的不对称性。主站要求具有较好性能，智能程度和复杂程度都相对较高，从站则性能要求较低。利用好这个特性对某些应用系统规划是非常有益的，大量从站可采用低成本的设计，从而有效提高系统整体的性能价格比。

主从轮询方案具有如下优点。

(1) 系统结构稳定而清晰，有利于通信协议的标准化；

(2) 由主站完全控制全局，不易出现控制失常而导致通信混乱；

(3) 协议可优化潜力大，可高效地利用信道资源；

(4) 方便增删计算机设备，只需设定主站即可完成添加和删除操作，该操作与大量从站完全无关。

主从轮询方案具有如下缺点。

(1) 系统存在明显的单点故障(Single-Point Fault，SPF)。若主站发生故障，整个网络系统将完全瘫痪。

(2) 协议具有非对称性，提高了协议机设计成本。

(3) 鉴于轮询机制，系统规模受到较大限制。大规模系统中对大量从站的轮询操作将严重降低运行效率。

主从轮询方案是一个有实际意义的方案，依据系统要求的不同可有多种技术上的变化。比较有代表性的主从控制协议是 HDLC。

2.1.4 令牌传递方案

借鉴中国古代将军排兵布阵时的做法，将一块令牌(token)授予任务执行官，取得令牌就等于获得授权，就有发号施令的资格。共享网络中也可通过令牌来授予和限定发送数据的权力。

定义一个共享网络系统中有唯一的令牌。当某个计算机设备握有令牌时，就拥有了发送数据的权限；当发送任务结束后，通过一定的令牌交换机制，把令牌传递给下一个设备。

令牌传递方法的基本算法相当简单，但实际情况下还是有许多有待谨慎处理的问题。

(1) 如何初始化令牌，让第一个设备握有令牌？

(2) 如何始终严格保持系统中令牌的唯一性？

(3) 如何发现令牌丢失？丢失后如何恢复？

(4) 如何处理系统故障造成出现两个或两个以上令牌的情况？

(5) 如何确定下一个令牌持有者？如果该设备恰好发生故障怎么办？

(6) 如何向系统中添加或删除成员?

上述问题都是设计协议时无法规避、不能忽略的,解决这些问题的过程就形成了协议中不可或缺的异常处理部分。虽然异常处理往往会很烦琐、很艰难,但无疑是保证网络系统在任何情况下都能够正常运行的必要条件。

令牌传递方案具有如下优点。

(1) 采用令牌控制发送权限,机制严密、规则清晰;

(2) 可以按需分配通信资源;

(3) 协议能够达到较高的执行效率,适用于高速网络。

令牌传递方案具有如下缺点。

(1) 有效控制和管理令牌是一项挑战,协议的复杂性程度较高;

(2) 增减站点将影响系统的其他计算机设备;

(3) 如果处理不当,令牌可能是系统的单点故障所在;

(4) 对信道质量要求比较高,光纤传输是最佳选择。

令牌传递方案以其算法的严密性而获得认可,实际运用该方案的有**令牌总线**(Token Bus)和**令牌环**(Token Ring)等网络(如图2.4所示),IEEE在局域网协议中制定的标准为802.4和802.5。

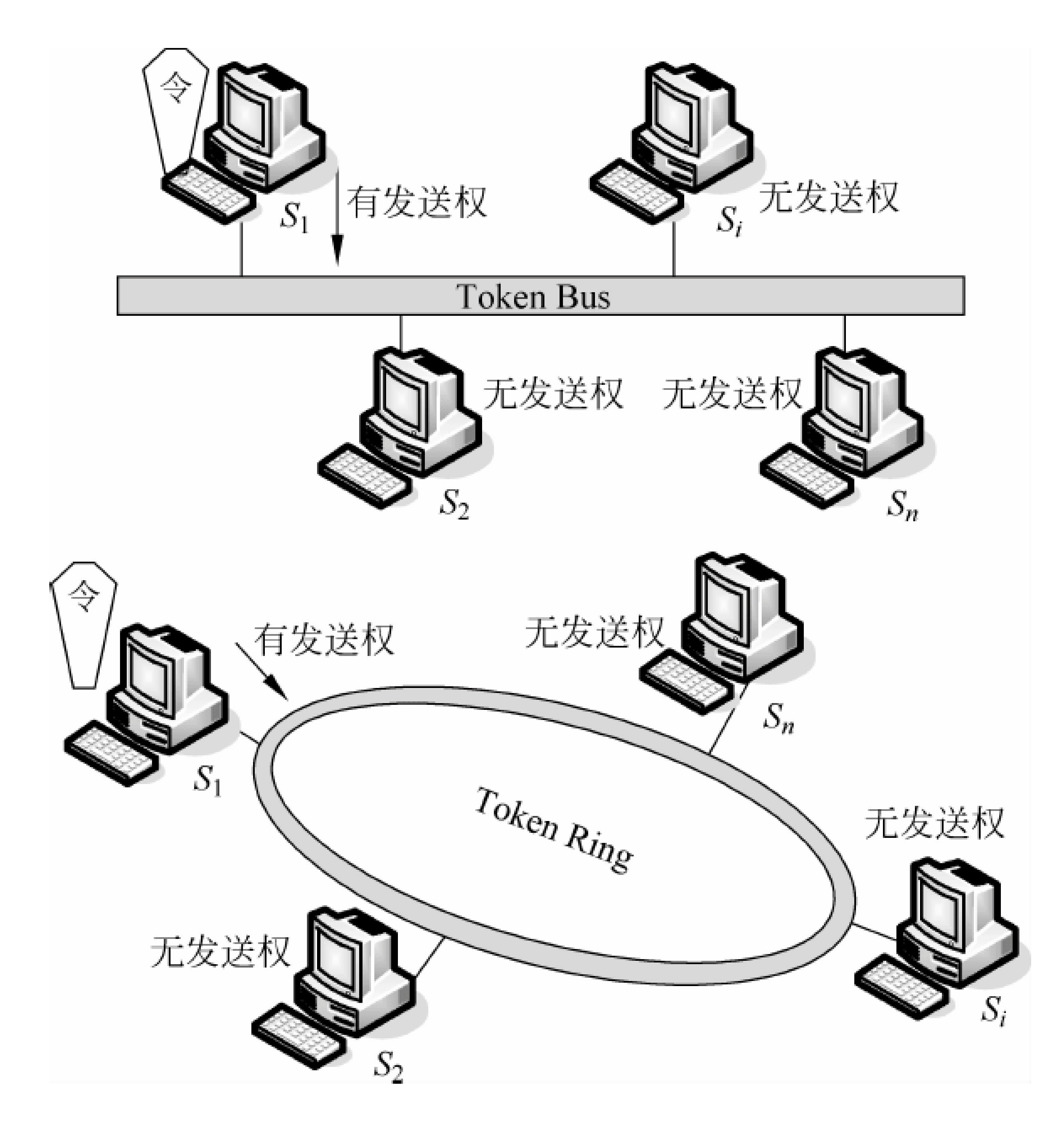


图2.4 令牌总线和令牌环网络示意

2.1.5 自由竞争方案

市场经济中的自由竞争,如果有法律法规的约束和公平透明的监管,是形成健康、繁荣的市场的基础。共享网络环境下,也可以让计算机设备自由竞争信道资源,如果采用合理的机制,就可用较小的代价达到简化管理成本、按需分配资源的目的。

如图 2.5 所示，简单的自由竞争方案为：有发送任务的计算机设备侦测共享信道，发现信道空闲，则启动发送；在发送的同时检测是否有碰撞（其他计算机设备同时启动了发送），如果发现冲突，则取消本次发送，重新侦测信道。

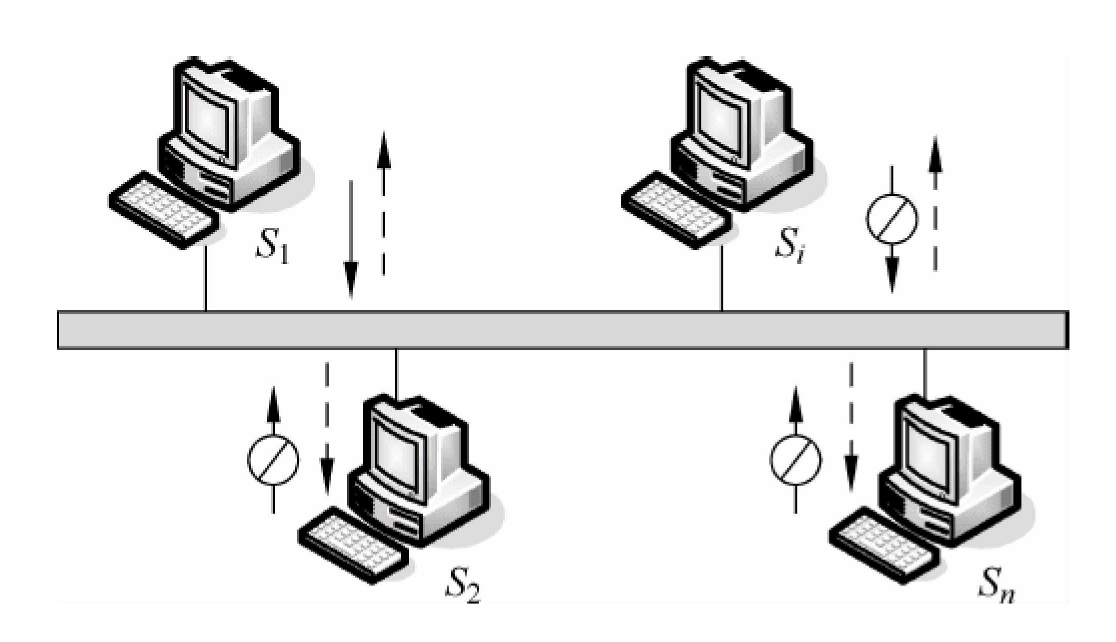


图 2.5　自由竞争共享总线示意

考察这一方案可以发现，自由竞争其实是一种有序的竞争，站点的发送行为具有自主性、自发性，然而所有站点都遵循特定规则，属于分布式控制技术。

自由竞争方案具有如下优点。

(1) 所有站点都是对等的（平等的），没有主从之分，每台计算机设备都采用完全相同的协议机，有利于标准化；

(2) 易于采用硬件和固件来实现算法，既提高了效率，又降低了成本；

(3) 加入和撤销计算机设备非常简单，可以随时地、在线地进行，不影响系统中其他设备。

自由竞争方案具有如下缺点。

(1) 网络规模受到一定限制。在共享网络中计算机设备数量达到一定程度后，碰撞概率提高，将导致发送成功率急剧降低甚至系统崩溃。

(2) 网络覆盖范围也具有一定限制。由于该方案采用信道状态侦测方法，随着信道距离不断延长，电信号的传输延迟就无法忽略不计了，这样，状态侦测的有效性就大打折扣，使碰撞不容易避免。正因为如此，该方案通常应用于以太网等局域网技术中。

2.1.6　带外信令方案

尝试跳出共享网络环境的条条框框，或许思路会变得更宽。基本想法是增加一套带外信令网络（Signalling Network），用以控制报文发送行为。

信令（signalling）是一种控制信号或控制报文，有别于承载用户信息的数据报文，用于管理呼叫建立、用户身份识别、呼叫拆除、计费信息等。信令网络专用于传递信令，是电信运营商网络的重要组成部分，为提供话务的电信级服务质量起到关键作用。如今逐渐把信令的概念和功能移植到计算机网络中。

带外（out-of-band）传输是指使用独立的网络来传输信息，不占用主体网络的带宽资源。相对而言，如果占用主体网络的带宽资源传输相关信息，就属于带内（in-band）传输。对于信令传输而言，通常需要区分带内传输方式还是带外传输方式，以评价信令传输是否会与主体网络产生相互干扰。

思考：共享网络环境下是否可以采用带内传输信令的方法？

如图 2.6 所示,共享网络带外信令方案的运作逻辑是:通过信令网络的协调,控制每个站点的报文发送动作,以保证同一时间只有一个设备在共享网络上发送数据。

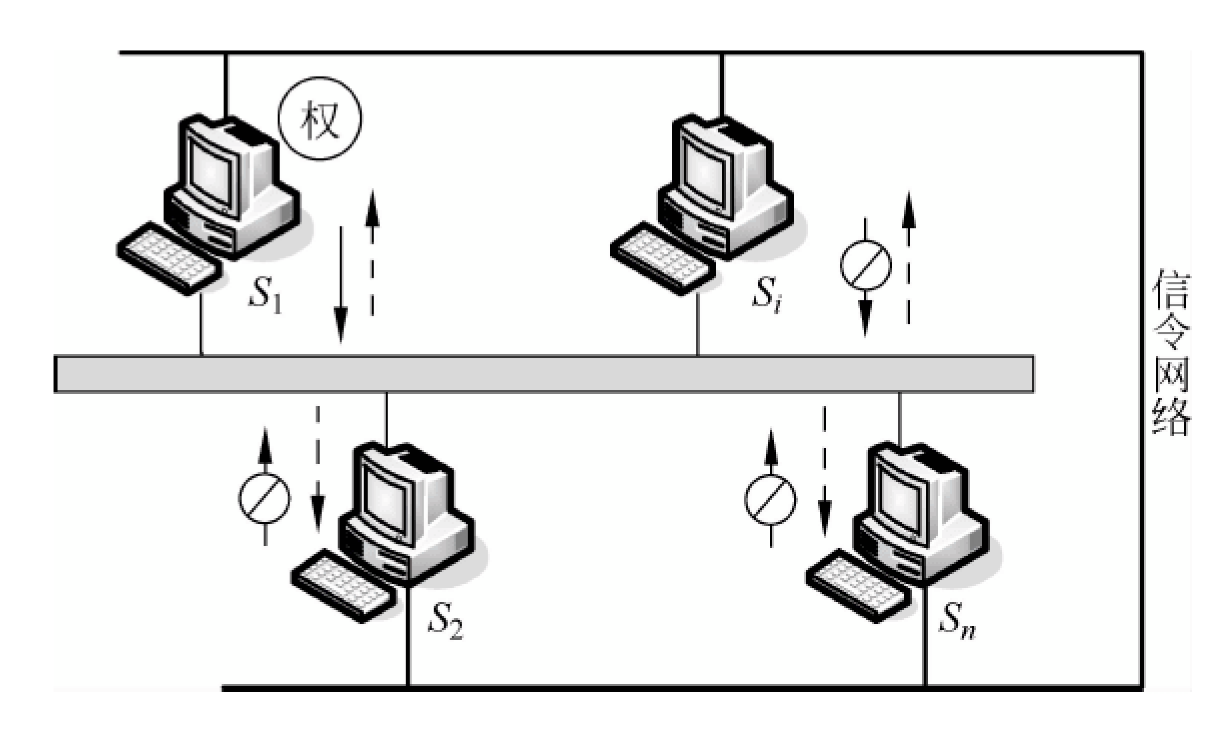


图 2.6　带外信令网络示意

带外信令方案具有如下优点。

(1) 控制逻辑清晰、有效,系统稳定性、可靠性较高,而且控制信息不占用网络有效带宽,提高了通信资源利用率;

(2) 所有发送设备地位均等,无主从属性之分,每台设备可采用相同的协议机实现,有利于标准化;

(3) 如果信令控制协议设计得当,设备增删不影响系统中的其他设备。

带外信令方案具有如下缺点。

(1) 独立的信令网络大大增加系统复杂性和成本;

(2) 一旦信令网络失效,系统将彻底瘫痪,即信令网络可能成为系统的单点故障源。

比较带外信令方案与时间同步方案,可以发现:时间同步方案是一种特殊的带外信令方式,因为时间就是独立于共享网络的控制机制。另外,令牌方案也等同于信令控制方法,但是没有设置信令网络,令牌就在主体网络中传输,应属于带内传输方式。

由此引出一个值得思考的问题:既然信令也是一种报文,那么,如果在共享网络中采用带内传输方式,是否本身也会发生冲突等问题?

事实上,类似于令牌方案的解决办法,带内信令是可行的,其实现原理和技术途径分析如下。

第一,时分(time-division)方法。初始时,共享网络传输信令报文,然后在信令的控制下进入数据传输状态,接着又返回信令传输状态,如此周而复始。

第二,背载(piggy-back)方法。在数据传输过程中,数据报文中附加了信令,该信令用以控制后续的传输过程。网络协议中常用的捎带确认机制就是运用了背载技术。

比较上述两个方法,前者的信令系统独立性较强,因此控制逻辑不容易因突发异常事件而发生混乱,后者的算法复杂性比较高,但通信效率也会相对较高。两者结合可以使信令控制系统更为完善。

2.2　Ethernet 协议原理

以太(Ether)是一种假想的空间电磁波传输介质,用于命名以太网,而以太网却是实实在在的存在。

以太网的本质是共享网络，技术核心是从 Aloha 协议发展而来的 CSMA/CD 算法，类似于 2.1.5 节讨论的自由竞争方案。如图 2.7 所示，以太网协议包括数据链路层和物理层，相关技术标准主要由 IEEE 802.1～IEEE 802.3 定义。

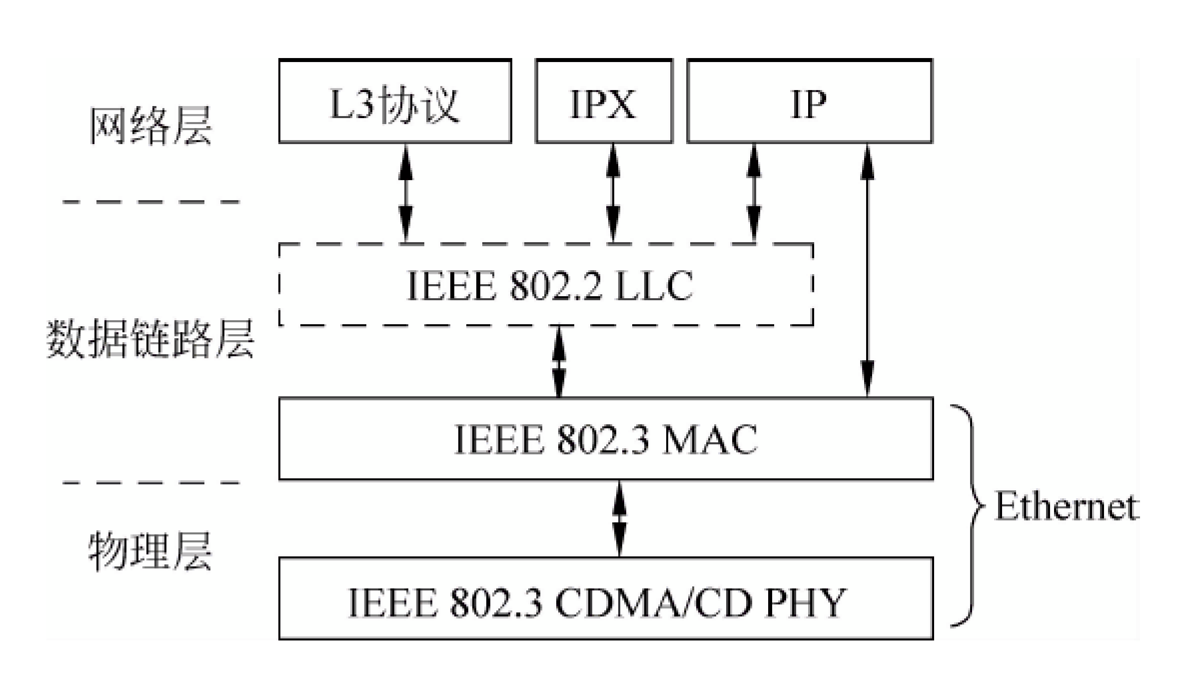


图 2.7　Ethernet 协议栈结构示意

2.2.1　Aloha 协议

Aloha 是夏威夷人相互打招呼的问候语，类似 Hi、Ciao、吃了吗。如果把计算机部署在夏威夷群岛的各个小岛上，那么计算机网络的作用就是把小岛连为一体，人们可以相互问候、相互通信。

共享信道通信技术的研究源自夏威夷大学，以 Aloha 来作为协议名称显得顺理成章。**Aloha 协议**主要面向无线信道通信，但同样适用于有线通信，其共性是共享网络通信方式，即任意一个结点都可使用信道说 Aloha，其他结点都可听得到，然而，当两个结点不约而同地说 Aloha 时，就会发生谁也听不清谁的情况，即产生信道争用（发送冲突）。Aloha 协议的核心目标正是解决冲突问题，使相互间友好的问候能够畅通无阻。

夏威夷大学的研究成果是成功地用无线信道互连了各个小岛上的计算机，让数字化的 Aloha 得以漂洋过海。Aloha 协议后来成为采用总线通信方式的 Ethernet 技术的鼻祖。

纯 Aloha 协议（Pure Aloha）采用随机接入方法。当结点有数据需要传送时，如果信道空闲，会立即启动发送；如果接收方正确收到数据，响应 ACK 确认；如果接收数据有误，则回复 NACK。当网络上的两个结点同时向信道发送数据的时候，冲突产生；两个结点发现碰撞后都终止发送，并停止一段随机时间，然后再次尝试传送。

纯 Aloha 难以应对过多的冲突，改进方法为**时隙 Aloha 协议**（Slotted Aloha）。时隙 Aloha 把信道在时间轴上进行分段，形成时间片，在满足发送条件的前提下，每个发送者只能在一个分段（时隙）的开始处进行传送，每次传送的数据必须少于或者等于一个时隙。这样在很大程度上避免了用户发送数据的随意性，减小了传输冲突发生的概率。

2.2.2　CSMA/CD 算法

带冲突检测的载波侦听多址访问（Carrier-Sense Multiple Access with Collision Detection，CSMA/CD）是 Ethernet 物理层采用的数据发送控制技术，用于共享媒介上多点设备通信，解决信道争用问题，并实现站点平等接入和分布式接入控制。

在同一条半双工总线上，为防止两台以上设备同时发送数据造成冲突，CSMA/CD 采用

载波侦听技术检测信道是否空闲或发生碰撞。如果一台连网设备从信道上可提取到曼彻斯特编码的信号，说明有其他设备正在占用信道，该设备就应规避发送；如果联网设备在发送信息的同时检测到信道的信号电压及其波动超过门限值(由多路电压叠加引起)，说明发生了碰撞现象，碰撞的报文无法被站点正常接收，本次发送失败。

理想情况下，碰撞一旦发生就会被所有站点发觉，但实际情况较为复杂，在高速传输中，存在感知滞后和先后等问题。为了保证 CSMA/CD 能够正确检测到冲突，应关注电信号的延迟，虽然延迟极其微小。

已知在电缆中电磁波传送速度约为 200m/μs，也即传送 1km 需耗时 5μs，则形成如图 2.8 所示的碰撞检测时序。

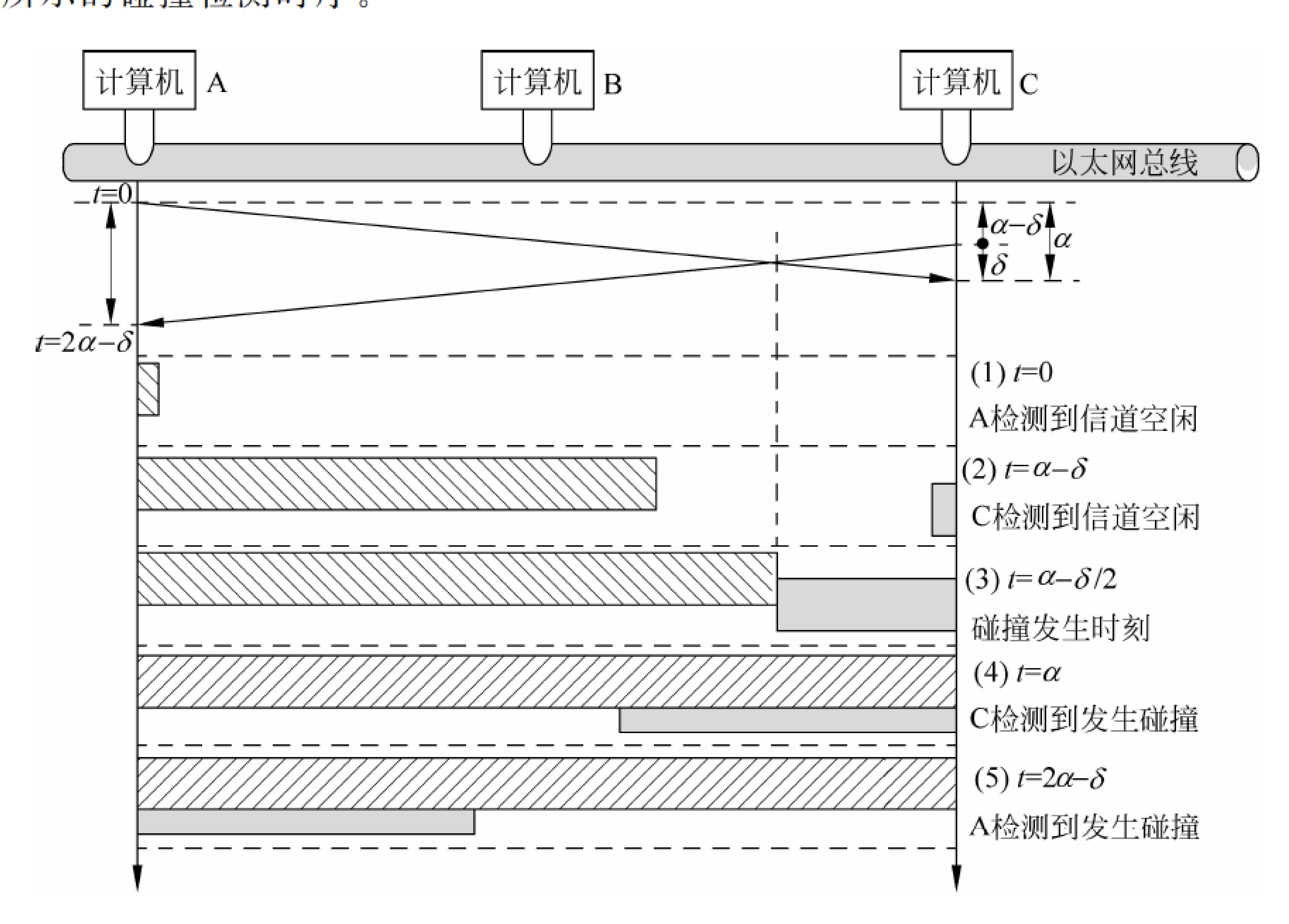


图 2.8　信号传播时延和碰撞时序

设相距最远的站点距离为 Sm，其间信号传播时延为 T_ds，则有

$$T_d = \frac{S}{200} \times 10^{-6}$$

又设报文长度为 Lb，带宽为 Bb/s，则报文传输时长 T_ps 为

$$T_p = \frac{L}{B}$$

根据图 2.8 所示的碰撞时序，最坏情况是当 A 发送的报文传送到 C 位置的瞬间($\delta \to 0$)，即 $t=\alpha$ 时，C 启动了发送，此时 A 的报文正好填满整个信道，如果 A 在这个时刻之前已经完成该报文发送(报文过短)，即 $T_p < T_d = \alpha$，那么，A 将因此无法了解发送的报文已经因为碰撞而报废。所以，应有 $T_p \geqslant T_d = \alpha$。

进一步地，从最坏情况的碰撞到 A 检测到碰撞，还需经过 α 时间，故对 A 的报文的约束还应该加强，应为 $T_p \geqslant 2T_d = 2\alpha$。则报文长度需满足

$$L \geqslant \frac{2 \times B \times S}{200} \times 10^{-6} = B \times S \times 10^{-8}$$

例如，当最远距离为 3km，带宽为 10Mb/s 时，有 $L \geqslant 300$b，即要求这种条件下最短报文长度应该在 38B 以上。考虑信道上信号中继器(如放大器)等产生的时延以及给予一定的宽限量，同等条件下应要求更大的最小报文长度，例如 64B。

这里 2α 称为**争用期**(Contention Period)或**碰撞窗口**(Collision Window)，是 CSMA/CD 必须关注的重要微观现象。

CSMA/CD 算法流程如下。

(1) 用载波侦听方法检测信道是否空闲。

① 若空闲，进入下一步。

② 若忙，等待并继续侦听信道。

(2) 发送数据，同时监测信道冲突。

① 若监测到冲突，进入下一步。

② 若成功发送完毕前未监测到发生冲突，成功完成，结束。

(3) 中止本次发送(发送一段干扰信号，告知其他设备碰撞已发生)，回收报文准备重发，进入下一步。

(4) 等待退避时间 T_w，返回第(1)步。

算法中退避时间 T_w 非常关键，否则，假如因碰撞而发送失败后立即转入重发，则两台(或多台)已发生冲突的站点将会同步进入下一轮发送，导致发生下一次碰撞的概率大大提高，很容易进入发送→碰撞→重新发送→再次碰撞的死循环。

CSMA/CD 采用截断二进制指数类型的退避算法来动态确定 T_w，以期降低再次碰撞发生的概率。

T_w 的取值方法为：

(1) 设 $T_0 = 2\alpha$(以太网实际取值 T_0 为 51.2μs)；

(2) 若重传次数为 R，取 $k = \min\{R, 10\}$(可知 R 越大则 k 越大，但最大为 10；当 $R \geqslant 16$ 后，取消该报文发送)；

(3) 设 $r \in \{0, 1, \cdots, 2^k - 1\}$，$r$ 为随机选取；

(4) 计算 $T_w = r \times T_0$。

2.2.3　MAC 协议

媒介访问控制(Medium Access Control，MAC)协议和**逻辑链路控制**(Logic Link Control，LLC)协议分别是 Ethernet 数据链路层的两个子层。

Ethernet V2 标准定义的 MAC 帧格式如图 2.9 所示。

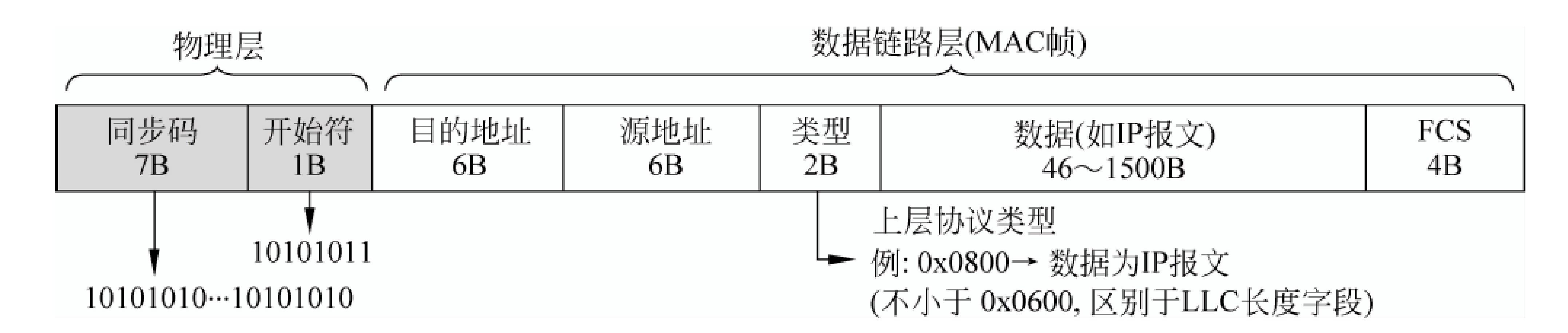


图 2.9　Ethernet V2 的 MAC 帧格式

MAC 帧中最主要的字段就是 MAC 地址,包括目的地址和源地址。MAC 地址由 6B(48b)构成,具有分段管理结构(如图 2.10 所示),用来唯一标识接入 Ethernet 的计算机(网卡)。IEEE 的注册管理委员会(Registration Authority Committee,RAC)负责全球的 MAC 地址管理,分配最高位的 3B(24b)机构唯一标识符(Organizationally Unique Identifier,OUI),或称公司标识符(company ID)。后 3B 为机构(生产商)自行分配的扩展标识符(extended ID),只需保证在自己的产品中唯一即可。由于 MAC 地址往往被固化在 Ethernet 网卡中,因此常被称做硬件地址、物理地址。

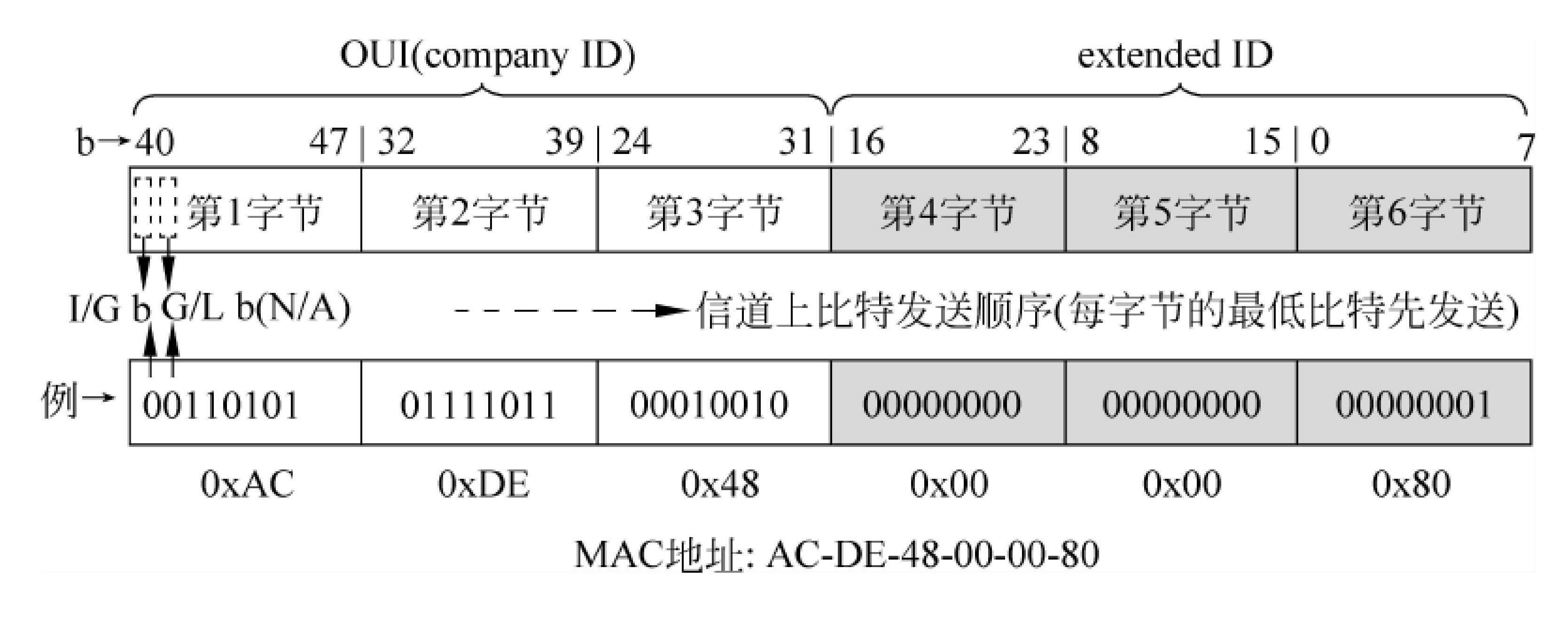


图 2.10　MAC 地址结构

此外,IEEE 规定,地址的第一个字节的最低比特为 I/G(Individual/Group),如图 2.10 所示,当 I/G=0 时,为单个站地址(相当于 IP 的单播地址);当 I/G=1 时,为组地址(相当于 IP 的组播地址)。当 MAC 地址为全 1 时,为广播地址。

还有一个几乎不使用的规定,第 1 字节的最低第 2 比特为 G/L(Global/Local),当 G/L=1 时,为全球管理;当 G/L=0 时,为本地管理。

在共享信道上,每一台计算机都可接收所有帧。Ethernet 网卡对接收到的每一个帧检查 MAC 目的地址,若为与本机 MAC 地址相符的单播帧、广播帧或可识别的多播帧,则接受该帧并进一步处理,否则丢弃之。

IEEE 802.2 定义了 LLC 子层,由 3B 帧头组成,插在 MAC 帧头之后的数据字段中(如图 2.11 所示),可见 LLC 子层在 MAC 子层之上。

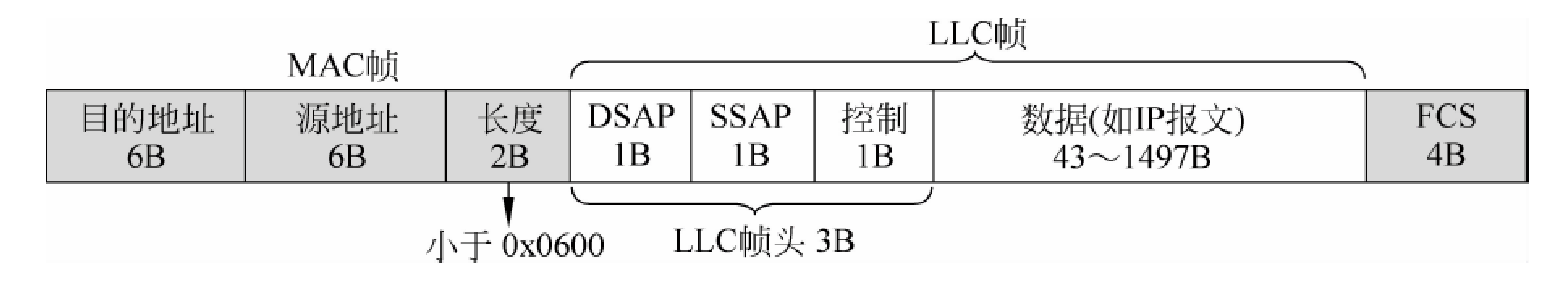


图 2.11　LLC 帧格式

DSAP 字段为 1B 的目的服务访问点,指出 LLC 的数据应递交的上层协议;SSAP 字段为 1B 的源服务访问点,指出发送数据的上层协议;控制字段为 LLC 帧的类型。但如今网络层几乎都使用 IP,支持上层多协议(如 IPX 等协议)的 LLC 已无应用价值,因此不再使用。

2.3　Ethernet 组网

以太网技术在多年的应用中不断地变化着、改进着、发展着，为自身的生存奠定了基础，也推动了应用能力的提高。

以太网的变化主要是在两个方面：带宽的变化，联网形式的变化。

带宽是体现网络能力的最为重要的指标。因此，带宽的提升贯穿了以太网技术发展的整个过程。从最初的 10Mb/s，到 100Mb/s 的快速以太网(Fast Ethernet，FE)，再到 1Gb/s 的千兆以太网(Giga-Ethernet，GE)，实现了数量级上的跨越，而且在相同媒介上是向下兼容的，最大限度地保护了已有投入。通信媒介也覆盖了双绞线、同轴电缆、光纤这三种有线通信的主要介质，甚至已经扩展到无线通信领域。这正是人们对以太网充满热情，充满信心、充满期待，也充满感激的原因所在。

但带宽的提高仅仅是一个方面，而且只是部分技术指标的改进。我们考察以太网技术的演变，其实从以太网联网形式的变化来看，或许更能说明问题，印象也可能更为深刻。

2.3.1　同轴电缆

采用同轴电缆(Coax Cable)来串接计算机，通信速率为 10Mb/s，采用 10Base2 细缆和 10Base5 粗缆，达到实现互连、形成网络的目的，是最初的以太网联网方式。从表面上看，计算机设备似乎是被串联起来的，而实际上却是并联的关系。

每台联网计算机都需要安装一块以太网网卡(NIC)，具有物理接口电路，用于完成接收和发送，包括数据编码和协议处理工作。

如图 2.12 所示，当计算机准备连接到以太网时，需要在同轴电缆上嵌入一个 T 形接头，在保持原有线路贯通的同时，把新的计算机设备并联到线路上。因此，所有连接到同一条同轴电缆的计算机将共同占有同一传输空间。

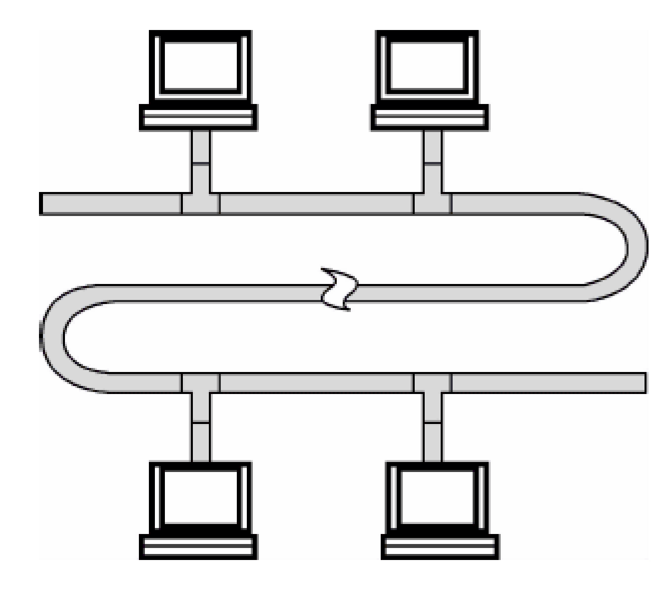


图 2.12　以太网同轴电缆联网方式

同轴电缆联网方式有其简洁、方便的优点。所有计算机设备，包括服务器和终端，都处于同等的地位，便于网络的部署和维护，增加和减少设备比较方便，甚至可以在系统运行中进行操作。单一设备的故障(除非是发送电路故障引起线路上信号紊乱)一般不会影响整体网络的正常运行，不会导致灾难蔓延。

但是，同轴电缆的联网方式还是存在明显的不足。

(1) 线路的布局受到较大制约，难以灵活调整。

(2) 增加设备时 T 形接头制作难度较大、要求较高，且容易损坏，还容易造成线路连接不良，严重的将造成断线。

(3) 同轴电缆成本较高。

(4) 受限于同轴电缆长度，网络覆盖的范围十分有限。

(5) 同轴电缆本身是单点故障易发点。

(6) 如图 2.13 所示，随着通信速率的提高，同轴电缆的通信效率会大大降低。

(7) 如图 2.14 所示,随着线路上设备数的增多,冲突增加,将导致网络通信效率急剧下降,最终可能引起瘫痪。

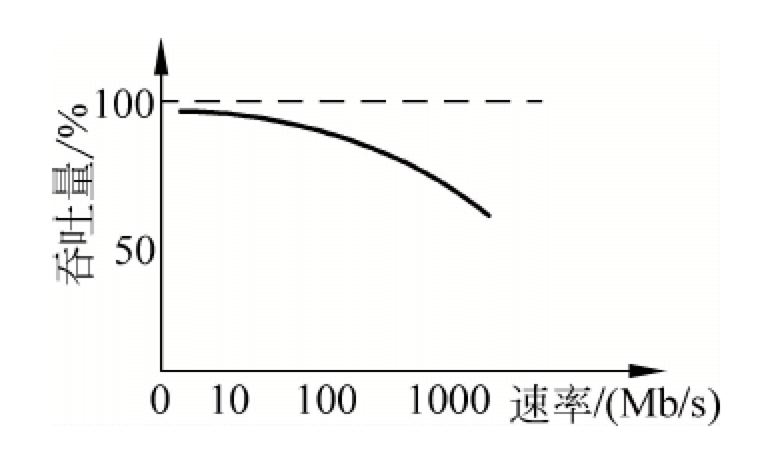


图 2.13 以太网吞吐量与速率关系

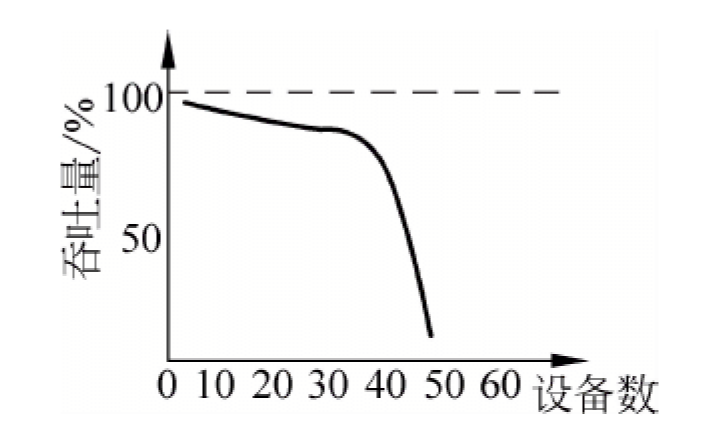


图 2.14 以太网吞吐量与设备数关系

吞吐量(throughput)是衡量网络通信效率的重要指标。吞吐量区别于速率或带宽,通常表现为一个百分比值,表示网络或线路的有效载荷。从严格的意义上说,吞吐量一定是小于 100%的,因为通信线路上的误码、校验位、分割符以及协议所需要的控制操作都会占用通信资源,造成有效数据的传送无法占据所有带宽资源。

以太网的同轴电缆组网技术虽然早已被弃置,但在网络发展初期起到了非常重要的作用,是后续技术标准的基础,并以此奠定了以太网在网络领域的领先地位。

2.3.2 集线器

集线器(hub)用于以太网组网,与同轴电缆连接方式相比,集线器组网技术存在两个显著的变化。

(1) 以太网集线器是技术飞跃式的创新,采用新的 10Base-T 标准。

(2) 通信介质使用非屏蔽双绞线(Un-shielded Twisted Pair,UTP),为 4 芯铜线,RJ45 标准接插件。

最常用的是五类非屏蔽双绞线(UTP5),为八芯铜线。在 RJ45 接口中:#1、#2 为橙色线组,#3、#6 为绿色线组,#4、#5 为蓝色线组(未用),以保持与 RJ11 接口的兼容,#7、#8 为棕色组(未用)。

从计算机端来看,#1、#2 线组为发送(Tx),#3、#6 线组为接收(Rx),从 hub 端看则反过来。因此计算机与 hub 的常规接线为(#1#2—#1#2,#3#6—#3#6),即通常所说的“直连线”,而交叉线接法为(#1#2—#3#6,#3#6—#1#2)。

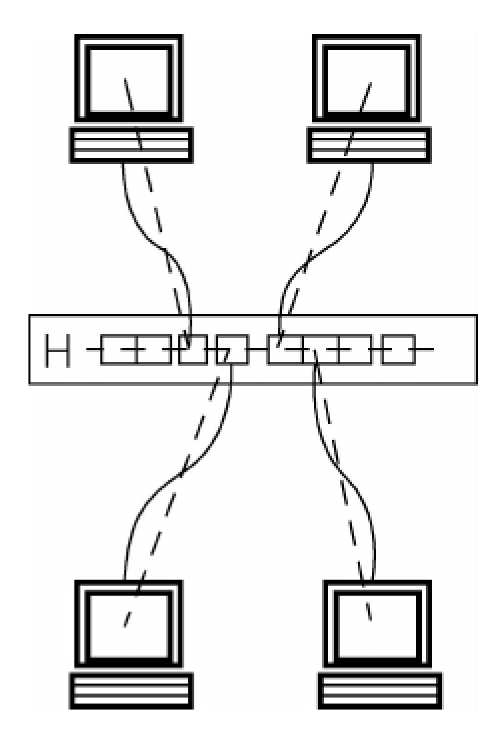


图 2.15 以太网集线器组网示意

集线器本质上是一台信号中继器。如图 2.15 所示,所有计算机设备都通过双绞线与集线器连接,形成星型的辐射状拓扑结构。每台计算机设备所发送的信息,都被集线器转接发送到所有端口,传输给其他计算机设备。所以,所有计算机设备实际上还是处在共享信道的环境中,即处于同一个冲突域,集线器仅仅进行了数据信号的无条件物理转发。

思考:*集线器联网方式的以太网属于何种拓扑结构?*

在以太网的集线器组网发展阶段,发生的变化不仅体现在连接手段上,还有一个进步也是非常重要的:通信速率从 10Mb/s 跨越整整一个数量级,升级为 100Mb/s,成为快速以太网。

一台集线器本身并不能称为网络，只有和计算机进行连接后，才能形成网络体系。集线器可以进行互连，以扩展端口数量，从而扩大联网规模。

相比同轴电缆联网方式，集线器联网方式获得了明显的进步。

(1) 支持全双工通信，在同一端口上发送和接收分离。

(2) 组网方式更为灵活多变，各台联网计算机的地理位置完全不受其他设备的制约。

(3) 容易进行较大范围的网络覆盖。

(4) 线路成本降低。

(5) 消灭了同轴电缆方式在线路问题上的单点故障。

思考：为什么说集线器方式消除了线路上的单点故障？

集线器的出现为以太网联网方式探索出一条新的途径。以往广域网联网才采用的联网设备技术在局域网中也得到应用。这种思路和技术也引出了以太网联网技术的下一步改进。

2.3.3　交换式集线器

集线器只是一种物理层的联网设备，有交换机的外形，却并不具有数据交换的功能。以太网交换式集线器(Switch Hub)才是真正意义上的数据交换机。

以太网交换式集线器的联网方式和集线器完全相同，目的也和集线器类似，仍然是转发数据报文，但数据交换机将对报文的目的 MAC 地址字段进行识别，并将报文转发给指定端口，而不会盲目地转发给其他端口。由于使用了以太网数据链路层 MAC 地址进行转发，所以也被称为二层交换机(Layer2 Switch)。

采用二层交换技术后，以太网 CSMA/CD 所固有的发送冲突被隔离开来，范围缩小到计算机与交换机端口之间；结合全双工通信技术，冲突问题就完全解决了。这是二层交换机给以太网技术带来的最有价值、最有意义的突破。

思考：二层交换机哪种情况下还是会出现冲突问题？

思考：二层交换机如何实现交换(基本工作原理)？

显然，排除了“不属于自己的报文”的无休止干扰，端口的利用率大大提高，网卡处理效率提升，数据安全性也得到一定程度的保障(防止窃听)。

二层交换机也可以相互连接而进行扩展，有两种连接方法。

(1) 级联(up-link)法：形成交换机的树状互连结构。

(2) 堆叠(stacking)法：使用特殊的专用高性能堆叠端口，相互组合，成为一台拥有更多端口的交换机。

思考：级联法和堆叠法的转发列表有何差别？在组网应用上有何差别？

在二层交换机基础上，通信媒介可扩展到光纤。同时，因为二层交换机拥有微处理器而具备计算能力，所以一些交换机提供了网络管理功能，可以进行远程配置、故障诊断、安全设置、数据统计等网络维护工作。

利用二层交换机，可以在网络中规划出特殊的子网，如 VLAN，将数据交换限定在 VLAN 范围内，内到外、外到内的数据交换都是受控的。VLAN 的划分限制了不安全的访问操作，并且可以防止第二层广播风暴(Broadcast Storm)的发生和蔓延。

2.3.4 三层交换机

三层交换机(Layer3 Switch)是由第三层协议参与报文交换过程,转发依据为网络层地址。由于如今采用的第三层协议均为 IP,因此三层交换机也称为 IP 交换机(IP Switch)。

IP 交换机组网与二层交换机组网方式类似。但三层交换机已经不再专属于以太网技术范畴了,因为以 IP 作为互连平台,交换机接口可以采用各种通信网络技术,除最常用的 Ethernet 外,还可采用 FR、ATM、MPLS 等。

2.4 WLAN

2.4.1 WLAN 体系结构

无线局域网(Wireless LAN,WLAN)即**无线以太网**(Wireless Ethernet),俗称 WiFi,由 IEEE 802.11 标准定义。

WLAN 网络结构如图 2.16 所示。

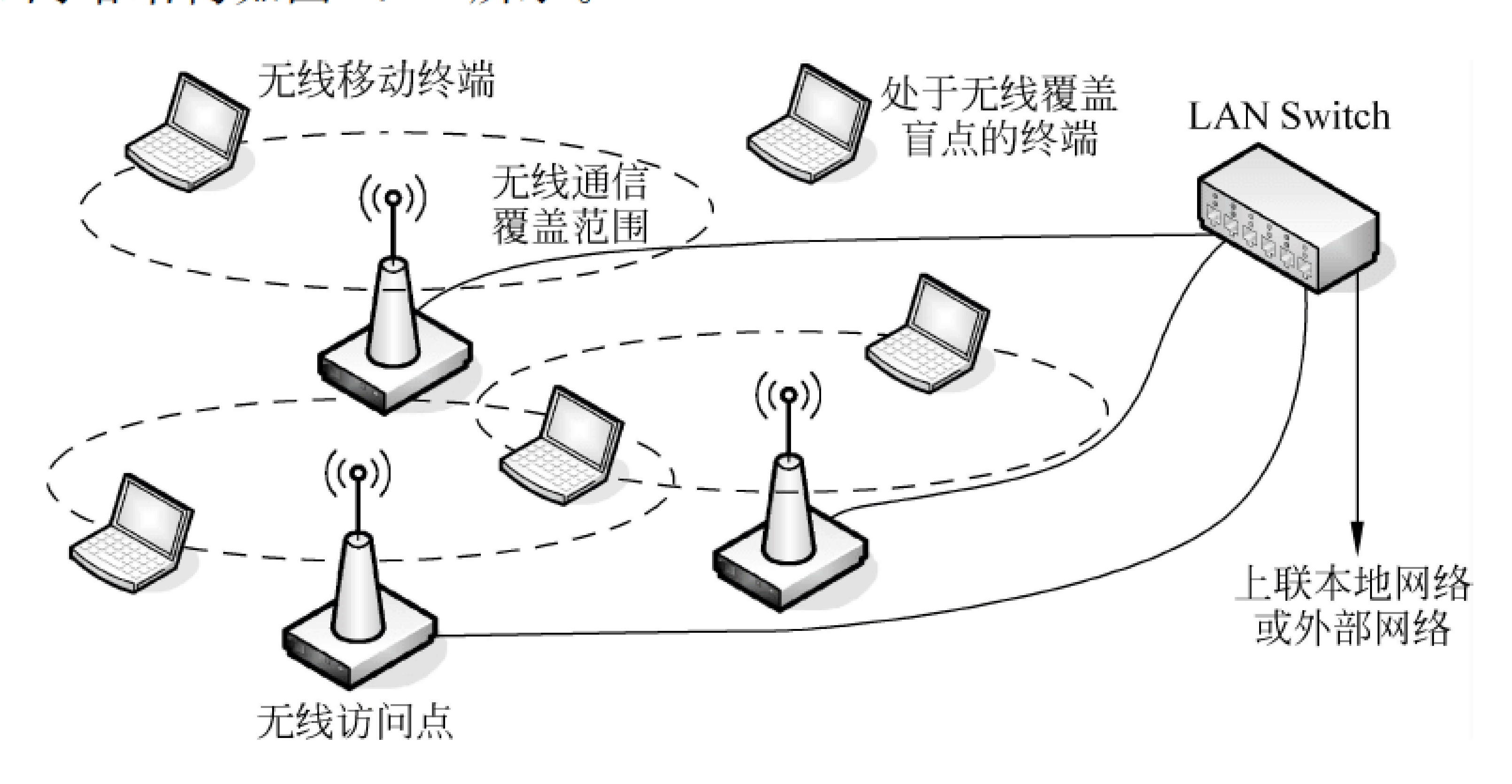


图 2.16 WLAN 网络结构示意

WLAN 主要有两种类型的设备:一种是无线终端,通常是计算机终端加上 WLAN 网卡构成;另一种称为**无线访问点**(Access Point,AP),其作用是提供无线信道和有线网络之间的桥接(bridge)。AP 由一个无线收发器和一个有线的网络接口(802.3 Ethernet)构成,桥接软件符合 802.1d 桥接协议。AP 就像是 WLAN 的一个无线基站,将多个无线终端聚合到有线网络上。无线终端可以是带 WLAN 接口的 PC、笔记本电脑、PDA 或手机。

WLAN 定义了两种工作模式(如图 2.17 所示):**基础设施**(Infrastructure)模式和**自组网络**(Ad-hoc)模式。

如图 2.17(a)所示,在常用的基础设施模式中,WLAN 至少配备一个 AP,与无线终端一起构成一个基本服务集合(Basic Service Set,BSS),覆盖一个基本服务区(Basic Service Area,BSA),是由 AP 的无线信号有效覆盖范围决定的。而一个扩展服务集合(Extended Service Set,ESS)是由两个或者多个 BSS 构成的,可以包括由门桥(portal)桥接的不同类型

WLAN。一般通过 WLAN 访问有线网络上的设备或服务(文件服务器、打印机、Internet 连接等)宜采用基础设施模式。

如图 2.17(b)所示,自组网络模式又可称为点对点模式、对等模式或独立基本服务集合(Independent Basic Service Set,IBSS),是一种无线网络特有的系统构成方式。以自组网络方式连接的无线终端之间直接进行通信,不需要经过 AP 或有线网络转接。这种方式可应用于不需要访问有线网络中的资源、只需要实现无线终端之间相互通信的环境。

无线终端接入 AP 需首先建立关联(association)关系,可检测 AP 周期性发送的信标帧(Beacon Frame)或采用主动发送探测请求帧(Probe Request Frame)来实现。终止关联称为分离(dissociation),而将关联转移到另一个 AP,则称为重建关联(reassociation)的过程。基于这种机制,可实现 WLAN 无线终端在 AP 间的漫游(roaming)。但漫游不等于无缝切换(Seamless Handover),后者要求网络连接及其业务不会因重建关联而中断或终止。

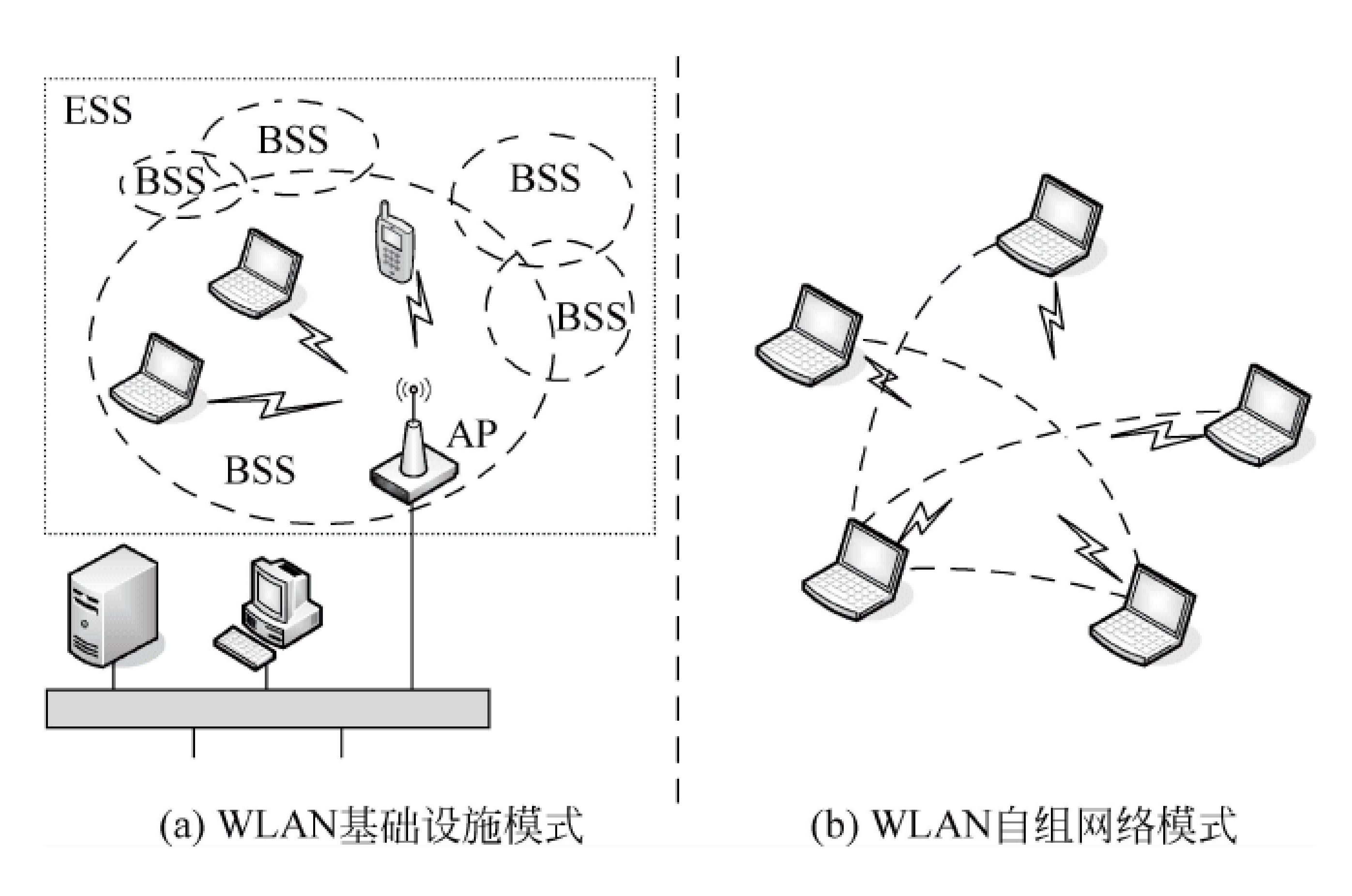


图 2.17　WLAN 两种工作模式

2.4.2　WLAN 物理层

WLAN 标准最初为 1997 年发布的 IEEE 802.11,包括物理层和数据链路层两个层次,后逐步扩展出 802.11a/b 等 18 种标准,其中 802.11a/b/g/n 是与物理层相关的标准,802.11d/e/h/i 是与 MAC 层相关的标准,802.11c/f 可增强 OSI 应用层的功能,802.11k/r 用于信道测量和快速切换,等等。

IEEE 802.11 的目标主要是解决局域网范围中用户终端的无线接入,采用面向分组的通信方式,对 IP 有良好支持,最高速率为 2Mb/s。802.11 物理层有三种实现方法。

(1) 跳频扩频(FH-SS)。使用 2.4GHz 的 ISM 频段(2.4000~2.4835GHz),共有 79 个频道供跳频使用(包括 4 个导频),第一个频道的中心频率为 2.402GHz,每隔 1MHz 一个信道。当使用二元高斯移频键控(GFSK)时,速率为 1Mb/s,当使用四元(GFSK)时,速率为 2Mb/s。

(2) 直接序列扩频(DS-SS)。也使用 2.4GHz 频段,但采用 14 个 22MHz 的通道(channel),临近的通道互相重叠,数据就是从这 14 个频段中的一个进行传送而不需要进行

频繁的跳频,避免额外开销。为了弥补特定频段中的噪音干扰,运用 Chipping 技术来解决这个问题。在每个通道中传输的数据都被转化成一个带冗余校验的 Chips 数据,与原始数据一起进行传输,用来提供错误校验和纠错,增加了网络的吞吐量。DS-SS 方法同样可提供 1 或 2Mb/s 速率。

(3) 红外线(InfraRed)。IR 波长为 850～950nm,速率为 1～2Mb/s,限于在室内环境使用。

2000 年 8 月发布的 802.11a 物理层工作在 5GHz 的 U-NII 频带,采用正交频分复用(OFDM)多载波调制技术,载波数达 52 个。物理层速率可选 6M、9M、12M、18M、24M、36M、48M 和 54Mb/s,可根据环境技术条件进行选择。802.11a 还可提供 25Mb/s 无线 ATM 接口、10Mb/s 以太网无线帧结构接口以及 TDD/TDMA 空中接口。

1999 年 9 月通过的 802.11b 是最常用的 WLAN 标准,物理层工作在 2.4GHz 频段,采用 DS-SS 扩频,编码技术为补码键控(CCK)和分组二进制卷积码(PBCC),调制方法为 DQPSK。802.11b 使用动态速率漂移,可因环境变化(噪音状况等),在 11M、5.5M、2M、1Mb/s 之间切换,且在 2M、1Mb/s 速率时与 802.11 兼容。

802.11g 标准是 802.11 b 标准的扩展(并兼容)。802.11g 在 802.11b 中添加了 802.11a OFDM 传输模式,拥有 802.11a 的吞吐率优势,但工作在 2.4GHz 频段。此外,802.11g 还定义了可选的增加吞吐率的模式。

802.11n 物理层采用了 MIMO 智能天线技术,将传输速率提高到 100Mb/s 以上。802.11n 使用 OFDM 改变数据包结构来兼容其他版本。

2.4.3 CSMA/CA 算法

如图 2.18 所示,802.11 的 MAC 层由点协调功能(Point Coordination Function,PCF)和分布协调功能(Distributed Coordination Function,DCF)组成。

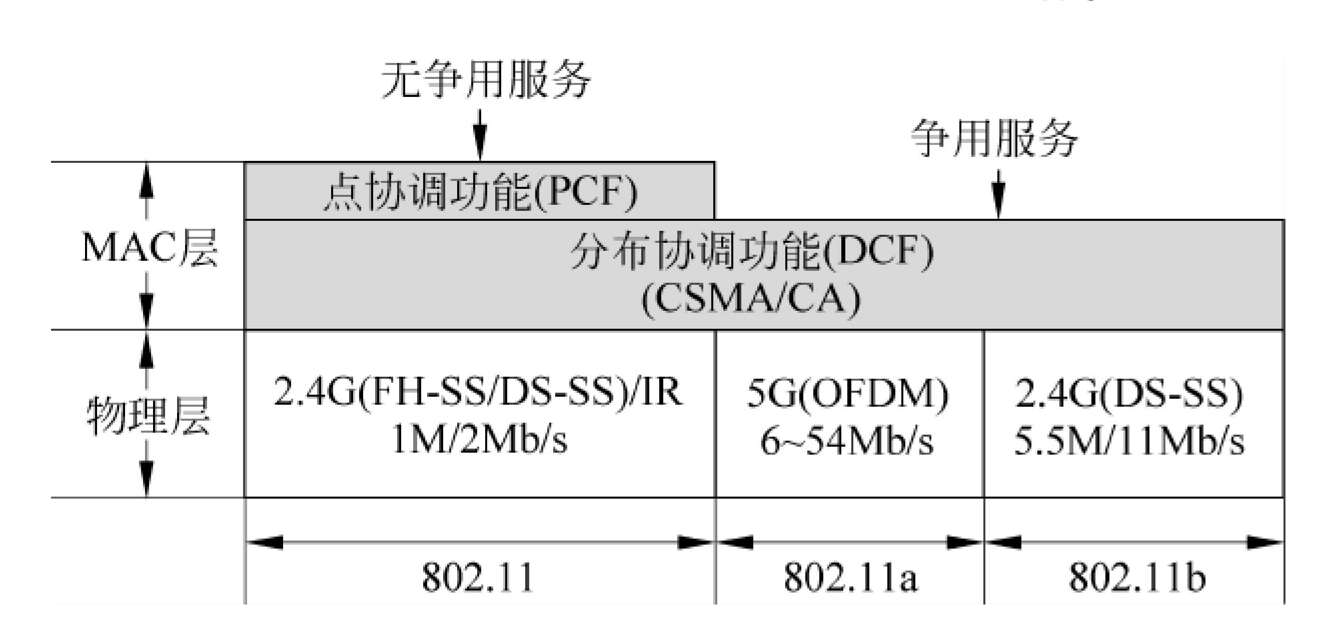


图 2.18　802.11MAC 层协议栈

协调功能用于确定在共享信道上无线终端是否可以发送信息,以避免冲突的发生。其中 DCF 执行 CSMA/CA 算法,而 PCF 为可选功能,使用类似轮询式的集中控制算法,将发送权轮流交给各个站点。对于时间敏感性业务,可使用无争用服务的 DCF,但自组网络没有 DCF 子层。

带冲突避免的载波侦听多址访问(Carrier Sense Multiple Access with Collision Avoidance,CSMA/CA)是 802.11 核心算法,与以太网 CSMA/CD 有相似之处,但考虑到无线信道的特性,不同点主要在冲突避免方面。

CSMA/CA 定义了一种**帧间间隔**(InterFrame Space,IFS)时间参数。其中 SIFS(Short IFS)最短,为 28μs,用于 ACK 帧、CTS 帧、由过长的 MAC 帧分片后的数据帧、所有回答 AP 探询的帧、在 PCF 方式中接入点(AP)发出的任何帧;PIFS(Point IFS)为 PFS 帧间间隔,比 SIFS 长,为 78μs;DIFS(Distributed IFS)为 DFS 帧间间隔,时间最长,为 128μs。三者分别相差一个时隙 50μs,即当某个站在一个时隙开始时接入到媒体,那么经过一个时隙时间,其他站都能检测出信道变为忙状态。规定一个站在发送完成后,必须等待某种 IFS 时间后才能继续监听并发送,并且高优先级帧等待时间较短,低优先级帧等待时间较长。

如图 2.19 所示,CSMA/CA 算法工作流程如下。

(1) 当站点有帧要发送时,先用载波侦听的方法检测信道,若收到的相对信号强度超过一定门限值,说明其他站正在占用信道发送。

(2) 若检测到信道空闲,在等待 DIFS 后进行发送(考虑高优先级帧可能要发送)。

(3) 目的站正确接收后,经过 SIFS,向源站发送 ACK 帧确认。

(4) 若源站在规定时间内没有收到 ACK(重传计时器超时),则重传该帧,直到收到 ACK 或超过重传次数后放弃发送。

(5) 当信道从忙变为空闲时,任何一个站先等待 DIFS 时间,然后进入争用窗口,等待随机退避时间,以减少碰撞概率。

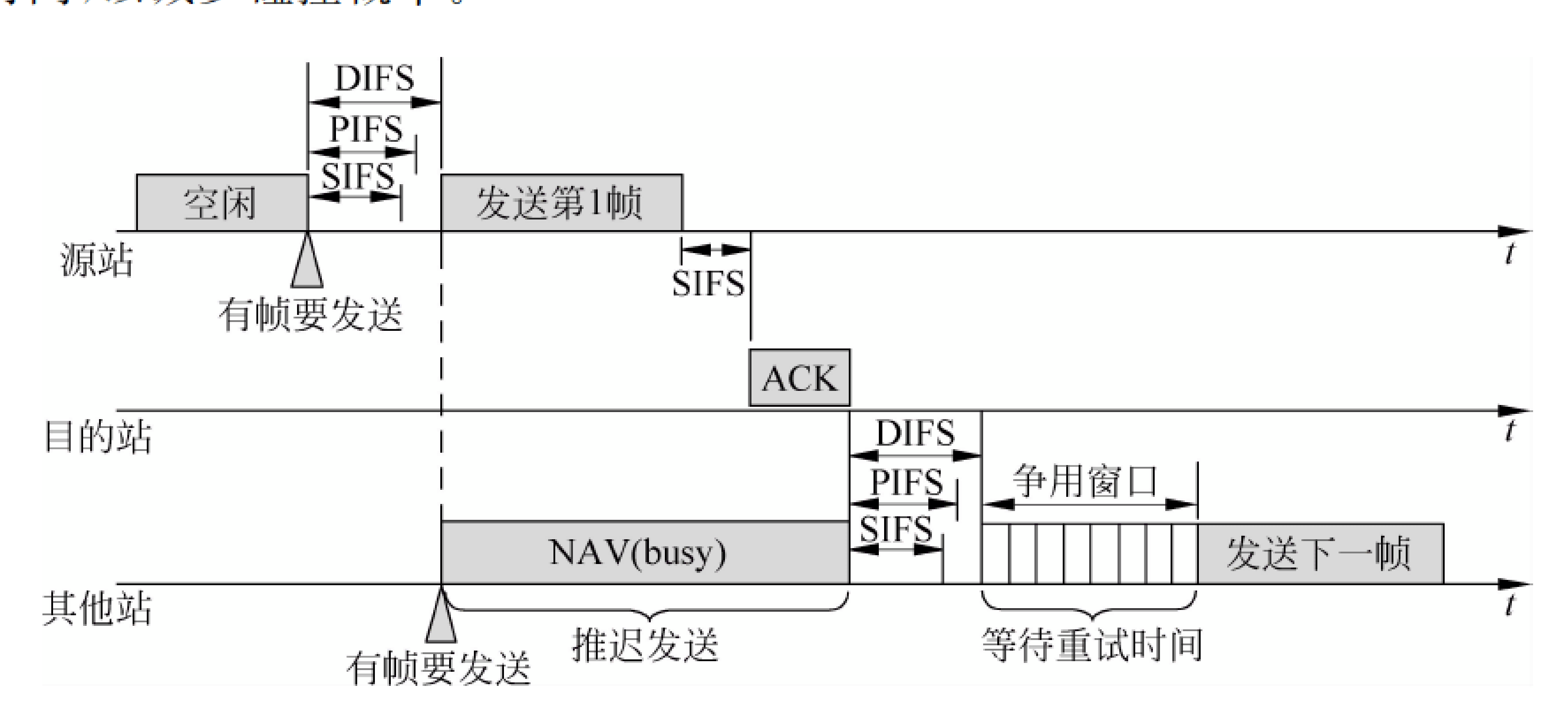


图 2.19　CSMA/CA 工作原理

CSMA/CA 采用退避机制的原因与 CSMA/CD 类似,随机退避算法也有相同的机理,但稍稍有所不同。CSMA/CA 退避算法是:第 i 次退避时,在[1, 2^{2+i}]中随机选择一个数值,作为应等待的时隙数。例如第 1 次退避时,应在 1～8 中随机选择时隙的个数。若退避计时器回零前信道变为忙态,则冻结计时器,等信道变为空闲后,再经过 DIFS,从原来的计时点开始继续计时。

CSMA/CA 还采用两种有效的机制来避免碰撞。

(1) 虚拟载波侦听机制。源站把将要占用信道的时间(包括目的站回复确认帧所需时间)通知给其他所有站,以便其他站在这一段时间都停止发送数据,称为**虚拟载波侦听**(Virtual Carrier Sense)。时间通知信息位于 802.11 MAC 帧首部(如图 2.20 所示)的第二个字段(持续时间字段),单位为 ns。

(2) 信道预约机制。为了解决隐蔽站问题,源站在发送数据帧前先发送一个专门的短控制帧(RTS),包括源地址、目的地址和预计将要占用信道时间(包括确认时间),若信道空

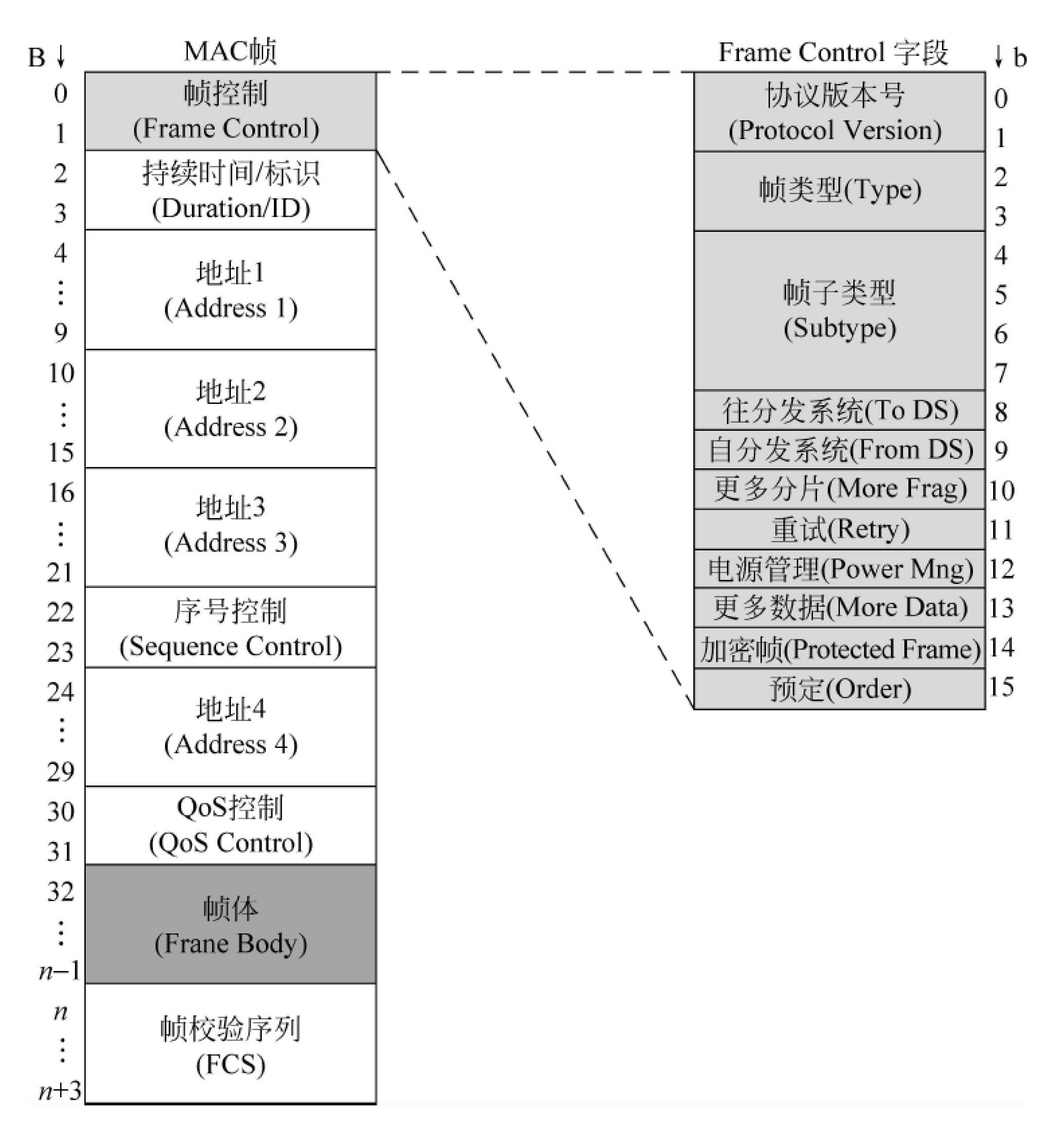


图 2.20　802.11 MAC 帧结构

闲,目的站就发送一个专门的响应控制帧(CTS),并复制 RTS 中的时间,源站收到 CTS 后就可发送数据帧。其他站不论收到 RTS 还是 CTS,都应推迟发送,推迟时间长度根据 RTS 或 CTS 中的指示,直到推迟时间到时或者发现目的站已经响应了 ACK,才能启动发送过程(如图 2.21 所示)。

思考:信道预约机制如何解决隐蔽站问题?能否解决暴露站问题?

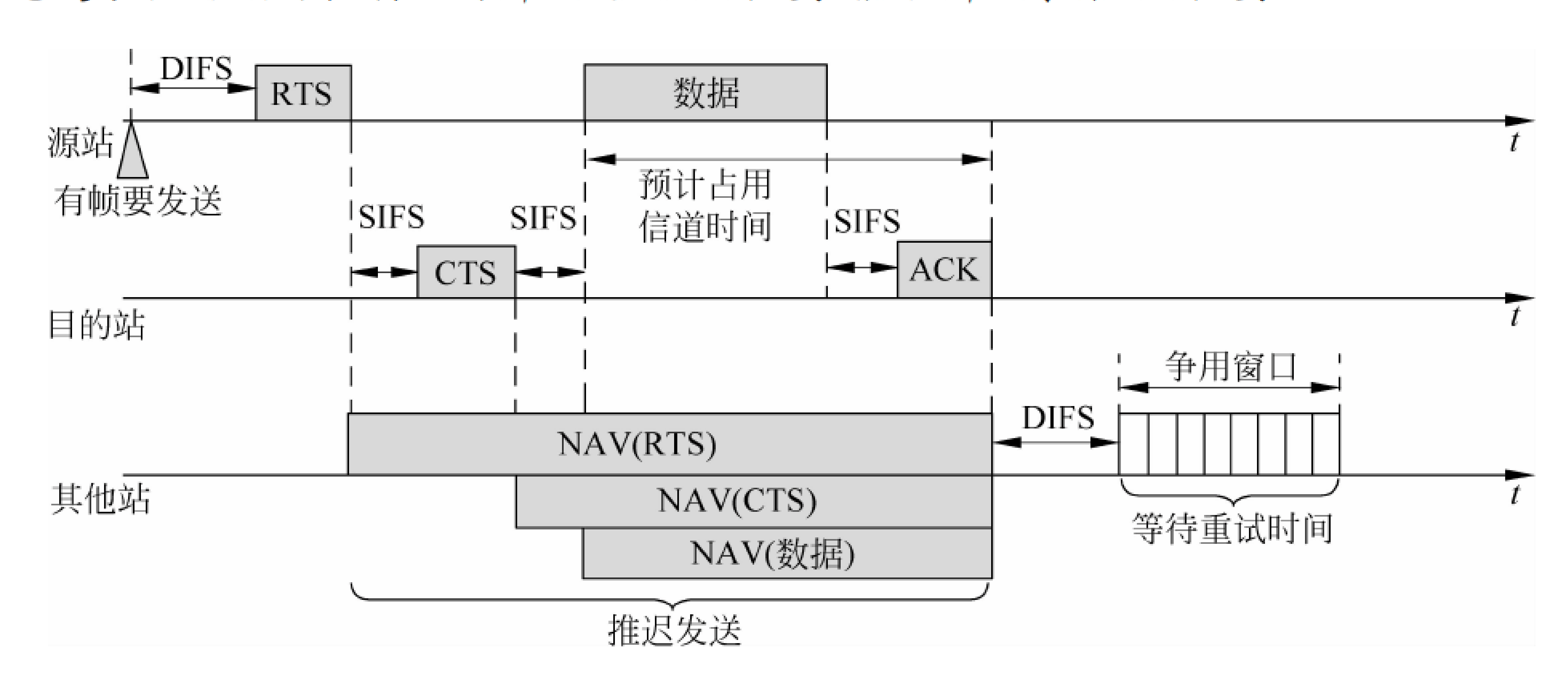


图 2.21　CSMA/CA 信道预约时序

2.4.4　WLAN 安全协议

1. WEP 协议

WEP(Wired Equivalent Privacy,有线等效保密)协议用于无线终端接入认证和数据加密,以防止非法用户窃听或侵入无线网络。WEP 是 IEEE 802.11 标准的一部分,使用密钥长度可变的 RC4 流式对称加密技术实现保密性,并使用 CRC-32 校验技术保障信息的完整性。

WEP 采用两种安全认证方式:开放系统认证(Open System Authentication)和共享密钥认证(Shared Key Authentication)。开放系统认证使用开放型的认证方式,只要输入密码,经验证正确即可获得通过。共享密钥认证方式采用以下四个步骤进行验证。

(1) 接入终端向接入点(AP)发送认证请求。

(2) AP 回复一个明文消息。

(3) 接入终端用密钥(设置的字符串或十六进制数值串)对明文进行加密,再次向接入点发送认证请求。

(4) 接入点采用相同密钥对加密数据进行解密,比较明文消息是否一致,并决定是否接受请求。

由于 WEP 在密钥和认证方面存在安全性上的不足,容易受到攻击,2003 年被 WPA 协议所取代。

2. WPA

WPA(Wi-Fi Protected Access,无线局域网保护接入)协议的作用与 WEP 类似,是对 WEP 安全技术的改进(过渡方法)。2004 年形成 IEEE 802.11i 标准(又称为 WPA2)。

WPA 的数据加密采用 TKIP(Temporary Key Integrity Protocol,临时密钥完整性协议)或 AES(Advanced Encryption Standard)技术,认证有两种模式可供选择:一种是采用 IEEE 802.1x 认证框架和可扩展认证协议(Extensible Authentication Protocol,EAP)的企业安全模式,另一种是称为预共享密钥(Pre-Shared Key,PSK)方式的个人安全模式,用于不需要认证服务器的家用或小型办公网络。

TKIP 仍然采用 WEP 所用的 RC4 加密算法,但加强了密钥的安全强度。TKIP 的密钥更长(128b),而且是动态变化的,即每个数据报文使用不同的密钥。其基本原理是:通过认证服务器或手工输入一个临时密钥用于会话,将临时密钥与每个站点的 MAC 地址进行混合,再加上 TKIP 序列计数器值、48b 初始化向量,产生 RC4 所用的加密密钥。

IEEE 802.1x 认证更为严格,需要配置专门的认证服务器,运行 RADIUS 或 Kerberos 协议,提供集中式安全认证和接入控制。

3. WAPI

无线局域网鉴别和保密基础设施(Wireless-LAN Authentication and Privacy Infrastructure,WAPI)是 2006 年由中国制定的无线局域网安全强制性标准(GB 15629.1101—1104),现已获得国际标准化机构的认可,与 WEP、WPA 等并列为 WLAN/WiFi 的安全保护技术。

基于 WAPI 协议的 WLAN 安全网络由无线终端、AP 和认证服务器(AS)三个实体组

成,运用公开密码体系完成终端和AP间的双向认证,认证过程中采用椭圆曲线加密算法(ECC,192b密钥),并协商生成会话密钥;通信过程中的数据加密采用国家密码主管部门指定的对称密钥加密算法(如128b密钥的SMS4)。WAPI支持在通信一定时间间隔或传输一定数量的数据包后,更新会话密钥。

WAPI主要包括以下两个方面。

(1) 无线局域网认证基础设施(WAI)不仅具有更加安全的鉴别机制、更加灵活的密钥管理技术,而且实现了整个基础网络的集中用户管理,从而能够满足更多用户和更复杂的安全性要求。

(2) 无线局域网保密基础设施(WPI)对MAC子层的MPDU进行加解密处理,分别用于WLAN设备的数字证书、密钥协商和传输数据的加解密,从而实现设备的身份鉴别、链路验证、访问控制和用户信息在无线传输状态下的加密保护。

CHAPTER 3

第3章 Internet 协议

Internet 又称**因特网**、**互联网**。自 20 世纪 70 年代起步，Internet 从一个小型实验网逐渐演变成为覆盖全球的超级网络，开创了网络时代和信息时代，使人类在享受真实生活乐趣的同时，可以自由自在地遨游于虚拟世界。

Internet 已经成为全世界的共同财富，凝聚了全人类的智慧。它是创新的平台、技术的摇篮、知识的宝库、发展的引擎。当我们为 Internet 陶醉和喝彩时，不妨重温以下这些里程碑式的历史瞬间。

(1) 1969 年 9 月 3 日，加州大学洛杉矶分校在实验室的两台计算机之间成功地进行了数据传输试验，ARPANET 诞生。麻省理工学院林肯实验室的拉里・罗伯茨为 ARPANET 项目主持人，被称为 ARPANET 之父。

(2) 1969 年 10 月 29 日，加州大学洛杉矶分校与斯坦福研究所实现首次网络连接。作为 Internet 前身的 ARPANET 初具雏形。

(3) 1972 年，雷・汤姆林森创立电子邮件应用，并采用@符号标记电子邮件地址，犹如神来之笔。

(4) 1973 年，ARPANET 建立了首个全球结点，位于英格兰和挪威。

(5) 1974 年，温顿・瑟夫与鲍勃・卡恩开发了 TCP，1983 年 1 月 1 日成为 RFC 标准。

(6) 1983 年，开始探索互联网域名系统(DNS)，一年后 com、gov、edu 域名启用。

(7) 1988 年 11 月 2 日，莫里斯蠕虫感染了 Internet 上数千台计算机。

(8) 1990 年，蒂姆・伯纳斯・李在欧洲核子研究中心开发了 WWW。

3.1 Internet 基本原理

Internet 的核心是 IP，因此 Internet 也称为 **IP 网络**。IP 为上层的 TCP/UDP 和各种应用层协议提供路由和寻址服务，并可以利用 Ethernet、ATM 等各种通信网络实现互联。

如图3.1所示,Internet协议栈包括网络层、运输层和应用层三个层次。可见,Internet并不关注数据通信所采用的技术手段是有线网络还是无线网络,是宽带传输还是窄带传输,是专线连接还是拨号连接,或者说,Internet可以兼容并蓄。其次,Internet没有OSI会话层和表示层协议(思考为什么),简化了协议栈,但仍然符合OSI模型。

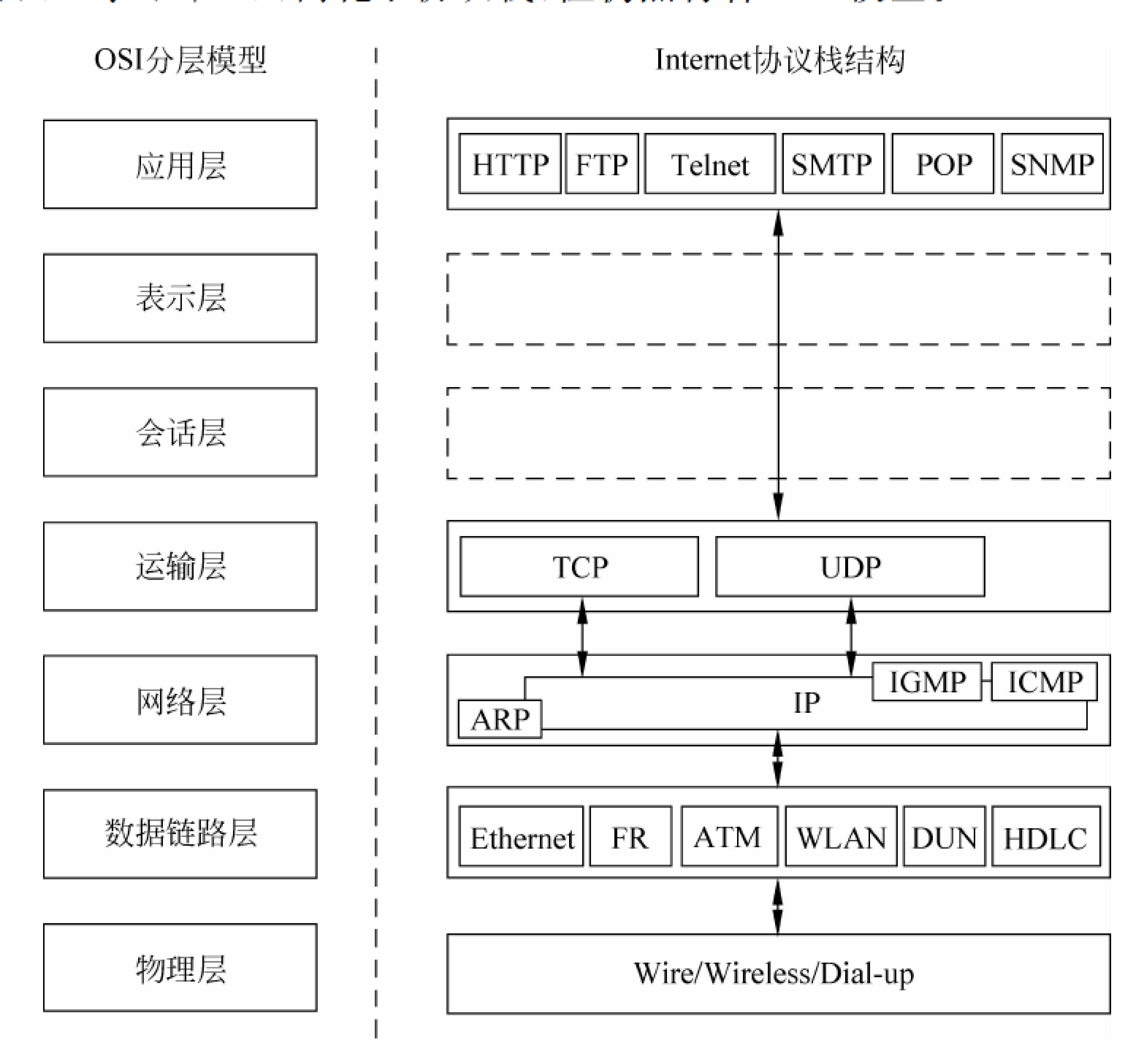


图3.1 Internet协议栈

虽然Internet非常庞大,其结构并不十分复杂,有很强的规律性。简而言之,Internet就是各种子网(subnet)的互联体。在符合Internet标准化体系的前提下,各个国家的IP网络通过主干线路相互连接,而各个国家IP网络同样是由地区性IP网络通过骨干网相互连接,层层递进,末端是用户子网,形成既有相对独立性又有相互关联性的互连体系,即成为Internet。

因此,Internet的网络体系很容易进行扩展,新成员的加入不仅不会影响其他成员的互联互通,而且使网络资源更加丰富。部分子网受损不会影响整体网络,不易引起灾害蔓延,具有出色的抗毁性。从技术上探讨,Internet网络的路由、寻址、互连是采用分布式控制方法,避免了集中式控制系统存在的性能瓶颈、单点故障。

Internet采用**路由器**(Router)互联各个子网。路由器运行特定的路由协议,承担连接不同网络、提供用户接入的任务。路由器根据路由信息,将输入的IP报文转发到目的IP地址相关的端口(线路),交给下一台路由器或子网,直到到达目标网络或终端。用户端的边缘路由器通常配备两个端口,一端连接用户子网,另一端连接Internet,因此并不需要依赖路由协议,仅需进行内网和外网的IP报文转发;网际互联路由器则往往配备多个端口,就应根据目的IP地址和路由信息决定转发的端口(即发送方向)。

Internet技术标准由IETF负责讨论和制订,以RFC标准文本的形式发布。常用的

RFC 标准如表 3.1 所示。

表 3.1 常用 RFC 标准

RFC 791	IP	Internet Protocol v4
RFC 1883	IPv6	Internet Protocol v6
RFC 792	ICMP	Internet Control Message Protocol
RFC 826	ARP	Address Resolution Protocol
RFC 903	RARP	Reversed Address Resolution Protocol
RFC 951/1541	DHCP	Dynamic Host Configuration Protocol
RFC 1034/1035	DNS	Domain Name Server
RFC 768	UDP	User Datagram Protocol
RFC 793	TCP	Transmission Control Protocol
RFC 821/822	SMTP	Simple Mail Transfer Protocol
RFC 1081/1939	POP3	Post Office Protocol v3
RFC 854/855	Telnet	Telnet Protocol
RFC 959	FTP	File Transfer Protocol
RFC 1350	TFTP	Trivial File Transfer Protocol
RFC 1866	HTMLv2	Hyper-Text Makeup Language v2
RFC 1945	HTTP 1.0	Hyper-Text Transfer Protocol v1.0
RFC 2616	HTTP 1.1	Hyper-Text Transfer Protocol v1.1
RFC 1157	SNMP	Simple Network Management Protocol v1
RFC 1001/1002	NetBIOS	NetBIOS
RFC 1005	SLIP	Serial Line IP
RFC 1661	PPP	Point-to-Point Protocol
RFC 1717	PPP-MP	PPP Multi-link Protocol

3.2 TCP/IP

3.2.1 IP

IP(Internet Protocol)是面向非连接的数据报协议(RFC 791),意味着 IP 不需要建立虚电路,不提供可靠数据传输。IP 为 TCP/UDP 传递 PDU,支持数据分割和重组;理论上,IP 报文可以通过不同的路径进行传送,在接收端进行按序交付。通常用 IP 指代 IPv4,即目前使用的版本 4 的 IP。

1. IP 报文

IP 只有一种报文(如图 3.2 所示),每个报文都携带完整的控制信息和数据,可以独立地在 Internet 上被传送(串行通信中,低比特优先发送)。

IP 报头(控制头)是一个 20～60B(必须为 4B 的整数倍长度)的结构化数据,常用 20B,可按照某些特殊系统的需要扩展协议选项字段(非 Internet 通用标准)。每个字段都有规定的位置、长度和含义,分别表示和执行相应的协议功能。

版本号(version)固定为 4,指使用 IPv4 协议。

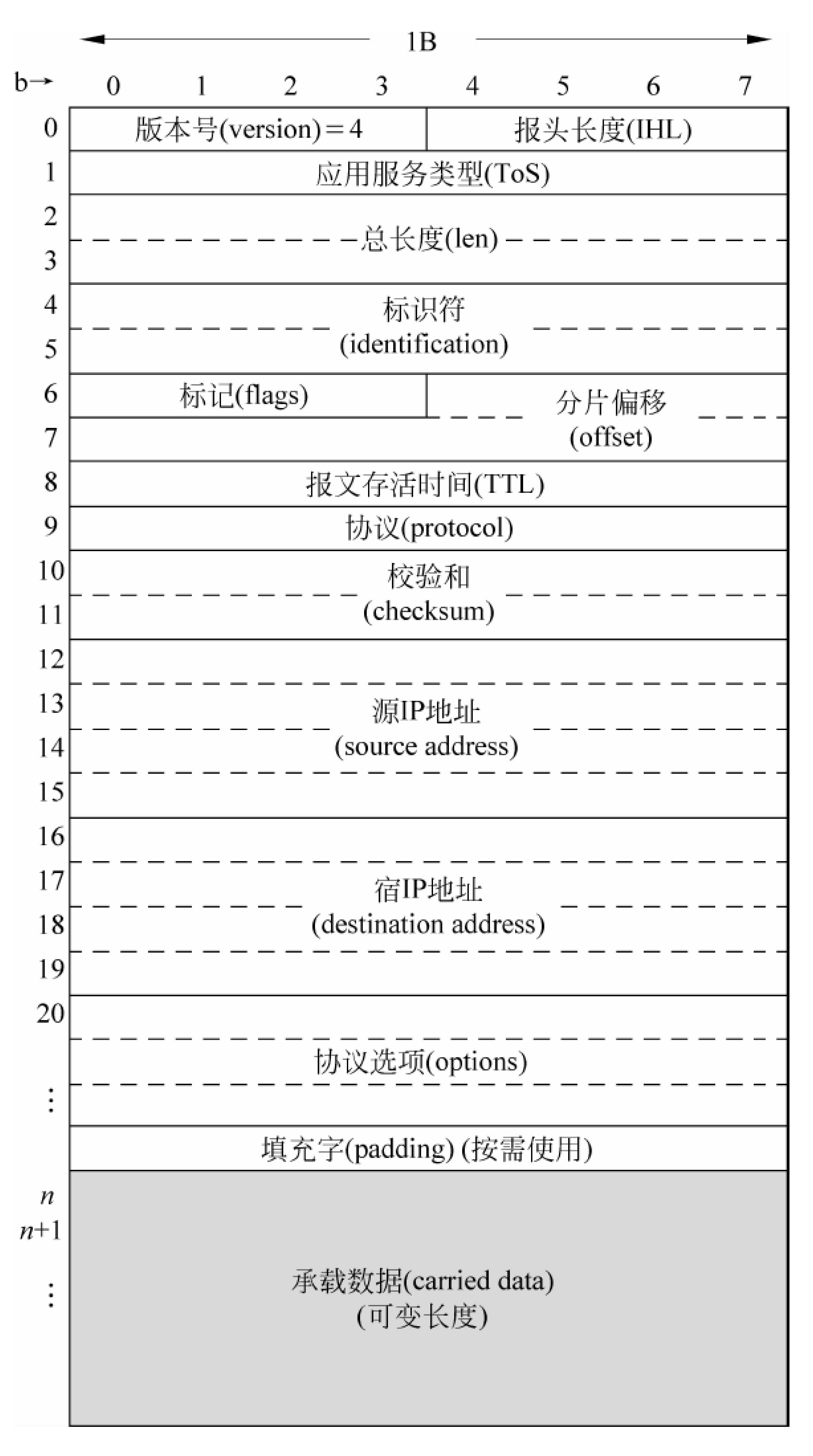


图 3.2　IP 报文

报头长度(IHL)是以 4B 为单位的数值,指明整个报头的长度。总长度(len)则为报头和承载数据的长度之和(以 B 为单位),因此(len-IHL×4)就是数据部分的长度。IP 报文的最大长度为 65 535B,但 IP 报文长度应当适应不同的数据链路层传输的要求,不能超过规定的最大传送单元(Maximum Transfer Unit,MTU)长度。例如,Ethernet 的 MTU 值为 1500,PPP 协议的 MTU 值为 296 等。

报文存活时间(Time To Live,TTL)是个非常有趣的参数,IP 报文居然也有寿命。事实上,TTL 非常必要。假如一个 IP 报文一直找不到目的地(例如由于传输差错引起报文变化),那么就可能在 Internet 上永无休止地兜圈子。这样的流浪报文积累得越来越多,就会导致网络效率严重下降甚至崩溃。因此,在一个 IP 报文生成的同时,会被赋予一个有限的 TTL,例如 16,这个 IP 报文每经过一个路由器结点,被称为一跳(hop,或称一跃),应将 TTL 寿命值减 1。如果报文在 TTL 归零之前到达目的地,没有任何影响;而一旦经过规定跳数(按最坏情况估算需经过的路由器数)之后还没有到达目的地,那么这个 IP 报文就会被

网络结点丢弃。

应用服务类型(Type of Service,ToS)指明所携带的数据属于何种类型,以便网络中继设备根据不同类型数据的要求作相应的配合。例如,文件数据要求网络保证可靠传输;语音信号必须尽可能连续,才能让对方听得明白,因此要保证语音报文优先得到传送的机会,不会间断;动态视频数据量巨大,丢失少量数据影响不大,但应尽量保证实时传送(短时延),避免因缓存和排队而堆积。这种按需处理数据的方式体现了非常重要的网络服务质量(QoS)要求。然而,Internet 网络对 ToS 几乎不处理,使这一功能形同虚设。

假如 IP 报文长度大于 MTU,就需要对 IP 报文进行分片(fragmentation),分段发送的 IP 报文应该在目的接收端能够拼接起来,以完全恢复原样。需要注意的是,分片不是简单地把 IP 报文切成若干段,而是指仅对数据字段进行分割(每一片数据长度须为 8B 的整数倍),报头部分仍然要完整有效。例如,设 IP 报头长度为 24B,数据部分长度为 3000B,准备在 MTU 为 296 的 PPP 网络上传送,则每个分段完成后的 IP 报文中数据部分长度为 296－24＝272B,那么,3000÷272＝11…8,即需要分为 12 个 IP 报文。

IP 报文分片功能使用报头中的 3 个字段:标识符(identification,ID)、标记(flags)和分片偏移(offset)。一次分片操作得到的所有 IP 报文应具有相同的标识符;标记字段最低位表示“还有分片”(More Fragment,MF),当 MF＝1 时,说明该 IP 报文后还有更多的分片报文,直到最后一个报文置为 MF＝0;标记字段中第二位表示“不能分片”(Don't Fragment,DF),DF＝0 时允许进行分片;分片偏移字段则指明该分片在原始报文中的相对位置,以 8B 为计量单位,第一个分片的偏移量为 0,如图 3.3 所示。

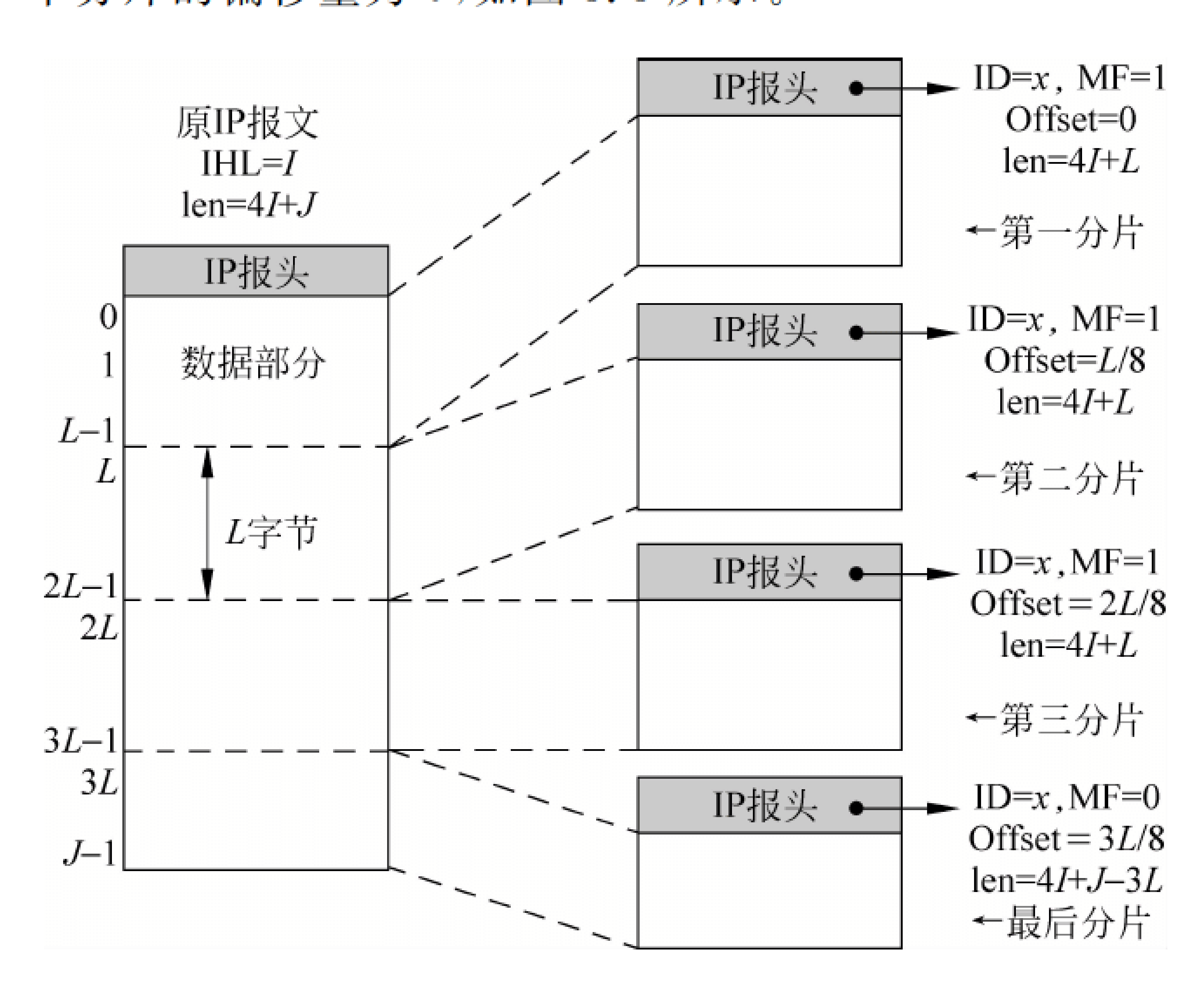


图 3.3　IP 报文分片示意

校验和(checksum)只对 IP 报文的报头进行校验。发送方先将校验和字段置 0,按 16b 字长进行反码算术求和运算,结果值的反码即为校验和。接收方校验时,同样进行反码求和,结果的反码如果为 0,表示数据报头未发生差错。如图 3.4 为 IP 报文发送方计算校验和的函数程序,其中巧妙地利用了反码求和的性质。

```
/* addr 指向 IP 报头首字节,count 是以字为单位的报头大小 */
unsigned short csum ( unsigned char * addr, int count )
{
  register long sum = 0;                  /* sum 存放校验和计算中间结果 */
  while (count -- )
sum += * (unsigned short) addr ++ ;       /* 先算术求和 */
/* 将位于高 16 位的进位数值加入低 16 位,重复直到无进位 */
  while (sum >> 16)
     sum = (sum & 0xffff) + (sum >> 16);
  return ~sum;                            /* 返回反码,即校验和 */
}
```

图 3.4　IP 报文校验和的计算函数

协议(protocol)字段指出了 IP 报文携带的协议类型,以便于目的方的 IP 协议机将数据部分交付给正确的上层(或子层)协议机。常用的协议字段值及其对应的协议如表 3.2 所示。

表 3.2　IP 报文协议字段常用值定义

协议分类	协议名称	协议字段值
运输层协议	TCP	6
	UDP	17
网络层管理协议	ICMP	1
	IGMP	2
网络层路由协议	EGP	8
	IGP	9
	OSPF	89
其他协议	IPv6	41

源 IP 地址(source IP address)和宿 IP 地址(destination IP address)都是 4B 字段,通常以十进制数表示为 w. x. y. z,例如:202.120.224.6。

协议选项(options)部分为可选的可变长度字段,最长为 40B,按需用全 0 的填充字段(padding)补齐为 4B 的整数倍。

2. IP 地址

IP 地址(IP Address)是 Internet 上计算机设备的通信标识,具有唯一性。

为了便于分配、管理和识别,IP 地址空间(0.0.0.0~255.255.255.255)被划分为 A、B、C、D、E 五类(如表 3.3 所示),地址第一字节的高位部分,运用霍夫曼编码进行区分及识别,分别用于不同规模的网络和不同用途。

A、B、C 类地址都具有网络号(net-id)和主机号(host-id)两个部分,而且都有公网地址和私有地址两种类型。公网地址也称为全网地址、全局地址、全球地址、合法地址;私有地址也称为内网地址、局部地址、保留地址、内部地址。主机号全 0 表示本网段,全 1 为本网段**广播地址**(Broadcast Address),否则就是**单播地址**(Unicast Address);D 类为**组播地址**(Multicast Address);E 类保留为**实验地址**(Experimental Address)。

表 3.3 IP 地址分类及其编码规则

	第 1 字节								第 2 字节								第 3 字节								第 4 字节							
A	0	S	S	S	S	S	S	S	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
B	1	0	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
C	1	1	0	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	H	H	H	H	H	H	H
D	1	1	1	0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
E	1	1	1	1	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*

注：S 为子网网络号(net-id)部分，H 为主机号(host-id)部分。

A 类地址(S. H. H. H)取值范围为 0.0.0.0～127.255.255.255，其中 10. H. H. H 为私有地址，127. H. H. H 保留给闭环测试用。易见，A 类地址共包含 128 个子网，每个子网有 $2^{24}-2=16\ 777\ 214$ 个主机(扣除本网地址和广播地址)。A 类地址适合同一子网中包含有大量主机的大型网络。

B 类地址(S. S. H. H)取值范围为 128.0.0.0～191.255.255.255，其中 172.16. H. H～172.31. H. H 为私有地址。B 类地址共有 16 384 个子网，每个子网 65 534 个主机。

C 类地址(S. S. S. H)取值范围为 192.0.0.0～223.255.255.255，其中 192.168.0. H～192.168.255. H 为私有地址。C 类地址有 2 097 152 个子网，每个子网 254 个主机。可见，C 类地址适合主机数量较少的小型网络。

D 类地址范围为 224.0.0.0～239.255.255.255。

E 类地址范围为 240.0.0.0～255.255.255.255。

IP 网络中采用**子网掩码**(subnet mask)与 IP 地址进行 AND(逻辑与)运算来提取子网地址(网络号)；显然，如果使用掩码的反码进行 AND 运算，就可提取到主机地址。A、B、C 类地址的子网掩码分别是 255.0.0.0、255.255.0.0、255.255.255.0。

子网掩码对于网络寻址和路由十分重要。如果两个 IP 地址使用子网掩码提取到的网络号相同，说明在同一个子网内，应采用二层以下的数据交换(如集线器或二层交换机)；如果一个 IP 地址的网络号与本网不同，说明属于其他子网，则应交付给路由器进行转发。

在实际应用中，子网掩码的应用可以非常灵活。子网掩码不一定是 8、16 或 24b 的 1，A 类地址的掩码也不一定是 255.0.0.0。

首先，可使用**可变长子网掩码**(Variable Length Subnet Mask，VLSM)，在主机号中划分出一部分编码空间用于网络号，可提高地址编码的利用率，提高网络效率并便于管理。例如，校园网中可选择使用 A 类私有地址 10. x. y. z，而子网掩码为 255.255.255.0，这样，可以划分出更多的子网($x|y$，最多达 65 536 个)，比如为每个系、每个教学楼分配不同的子网，有利于网络维护。

VLSM 方法对于提高路由效率有重要的作用。如图 3.5 所示，从一个结点出发到达其他结点，a 方式下需要记录和维护到达所有结点的路由信息，而在 b 的子网方式下，只需维护本子网结点的路由信息，其他结点一律由网关“接力”到达，有效缩短了路由表的长度。

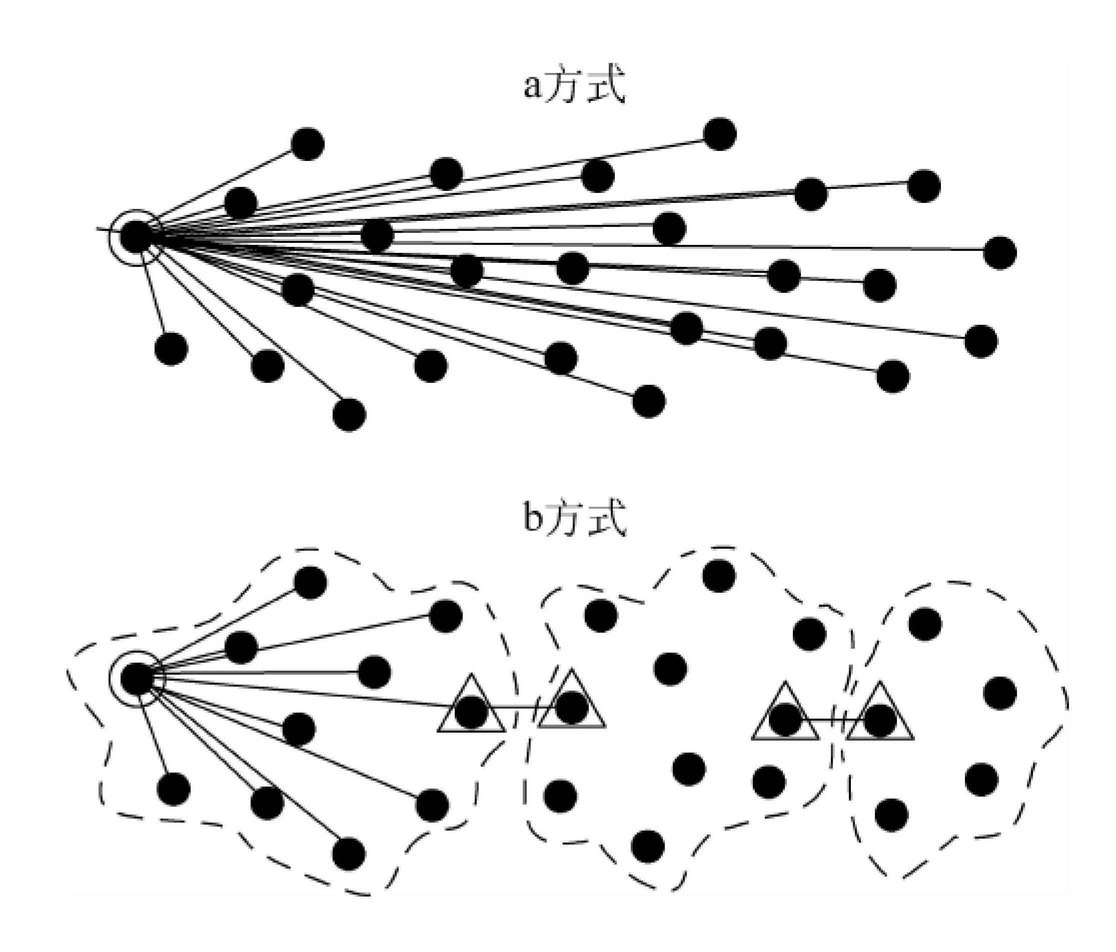


图 3.5　全网路由与子网路由

其次，在路由协议（尤其是外部网关路由协议）中，可根据网络分布情况，将同向转发的相邻的子网地址合并起来，成为**超网**(super-net)，可以大大简化路由表，提高路由效率。用于表示超网的方法就是**无分类域间路由选择**(Classless Inter-Domain Routing，CIDR)。

如果说 VLSM 可将一个大子网"分拆"为若干个小子网，那么，CIDR 则着眼于"聚合"。当然，CIDR 并非将子网合为一体，而是对子网地址的汇聚，使得网间路由信息更为简洁。与 VLSM 一样，CIDR 打破了以 A、B、C 类固定划分子网的僵硬机制。CIDR 用网络前缀(network-prefix)来代替网络号、子网号和主机号，采用"网络前缀/前缀长度"的结构形式。

例如，如图 3.6 所示，有 16 个连续的 C 类地址，最小地址为 192.14.32.0，最大地址为 192.14.47.255，地址的高 20b 相同，即可用 CIDR 合成为一个大地址块，记为 192.14.32.0/20，可省略低位连续的十进制 0 简写为 192.14.32/20，其中 20 表示前缀长度为 20 个二进制位，所含的地址总数为 2^{12} 个。

IP 地址中，公网地址需要向 ICANN(the Internet Corporation for Assigned Names and Numbers，互联网名称与地址分配机构)或其代理机构申请获得，具有全网唯一性，可在 Internet 上被访问到，因而成为用一个少一个的稀缺资源。私有地址则无须申请，可以自由分配使用，但仅限于在内部网络范围内有效，无法在 Internet 上实现寻址操作。如果一台使用私有地址的主机需要访问 Internet 网站，则必须使用 NAT 方法转换为公网地址。

IP 地址是 Internet 上计算机设备的标识，但地址编码难以记忆、输入烦琐，为此，Internet 设计了**域名**(Domain Name)表示法，以接近自然语言的方式来标记计算机，好比手机上的电话号码本，可以让访问者不必背诵冷冰冰的长串数字，而是用亲切好记的名字。而且，即使 IP 地址变了（例如网站迁移），只要域名不变，就不会流失忠实用户。

域名采用点分字符串结构，允许字母（不区分大小写）、数字和连字符(-)的组合，构成方式如下：

[二级域名].注册域名.顶级域名

最初，Internet 定义了六个顶级域名：com、edu、org、net、gov、mil，用以区别不同的行业领域；随着 Internet 推广到全球，域名的应用也越来越普及，先后推出了区域性顶级域名（如 com.cn、net.hk）和国家顶级域名（如 cn、jp）；之后还扩展出中文等非英语字符的顶级域名（如中国）。

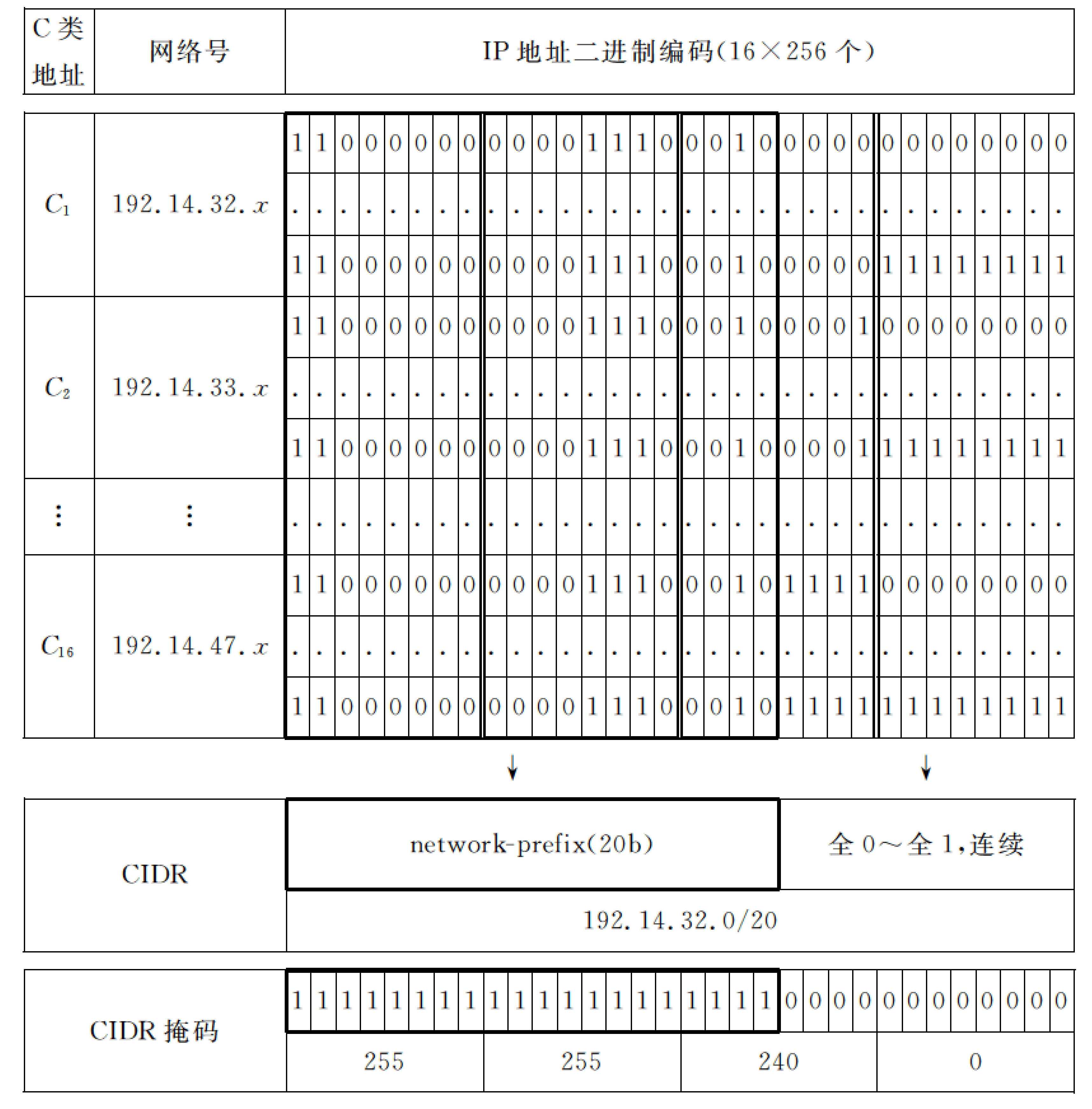


图 3.6　CIDR 地址聚合实例

特定顶级域名下的注册域名必须向域名管理机构(如 ICANN、CNNIC)申请获得,具有唯一性。由于采用先到先得的原则,域名成为了 Internet 上的宝贵资源之一。

思考:为什么有人愿意花钱去抢注、囤积域名?

二级域名是由用户在注册域名下自行分配的,用于管理信息系统中的不同体系,例如大型网站的不同频道、企业网络的不同应用。

一个域名对应于一个 IP 地址(IP 地址允许改变),一个 IP 地址可以绑定一个或多个域名。但 IP 只能识别 IP 地址,不能直接使用域名,因此,Internet 提供了将域名映射为 IP 地址的域名服务系统(Domain Name Service/System,DNS),实现域名解析。

DNS 于 1983 年由保罗·莫卡派屈斯(Paul Mockapetris)发明,发布为 RFC 882 标准,1987 年修正为 RFC 1034/1035。如图 3.7 所示,DNS 具有类似目录树的等级结构,分主服务器和转发服务器,采用客户机/服务器访问模式。当用户在应用程序中输入域名后,由计算机向指定的主 DNS 或辅 DNS 服务器(事先已由人工或 DHCP 设定)发出解析请求;若 DNS 服务器检索到域名,则立即响应该请求;若 DNS 服务器检索不到该域名,则向其上一级 DNS 服务器发出查询请求,而根 DNS 服务器也可向管理该域名的其他区域 DNS 服务器进行查询,类似一个递归查询过程,直到解析出域名,然后层层回复,最终交付给需求方。沿途的 DNS 服务器可以缓存获得的域名解析结果,以便下次查询时可以直接回答。缓存方法虽然可以提高 DNS 效率,但域名归属地所做的 IP 地址变化不能立即反映出来,可能引起解

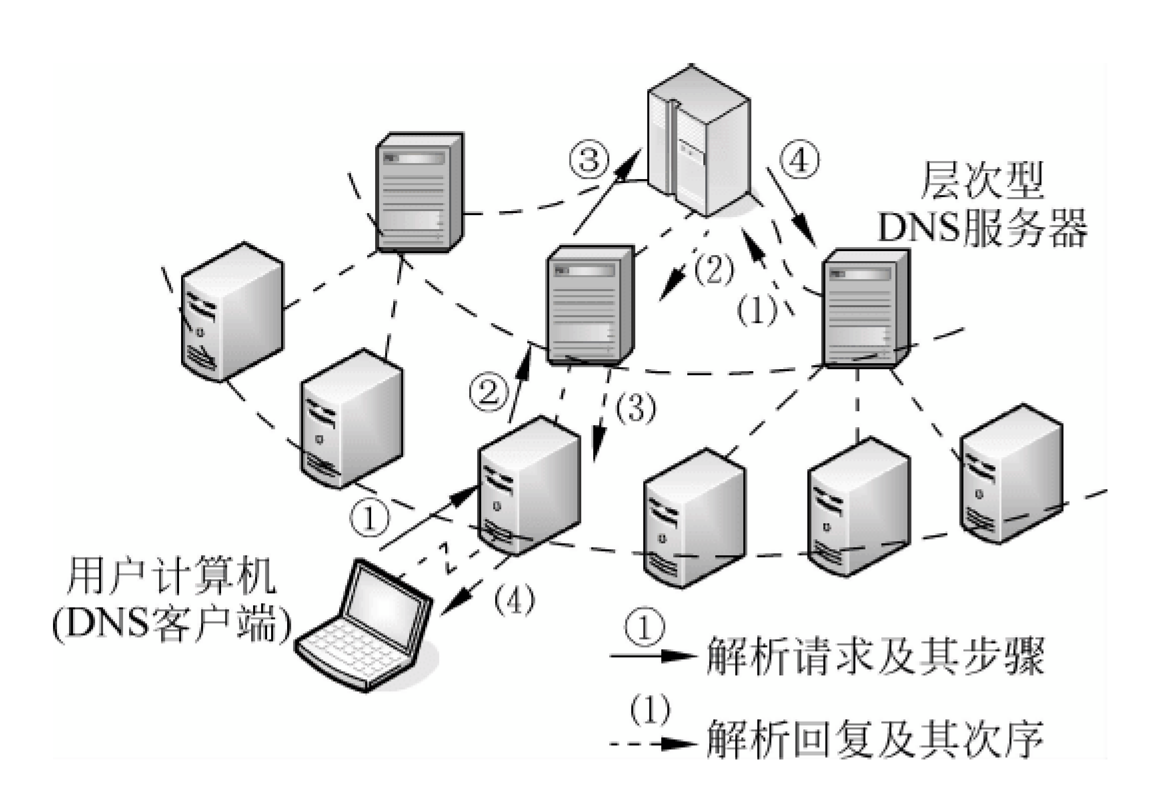


图 3.7　DNS 体系结构原理

析出错而导致访问失败。

DNS 使 Internet 访问更便利,然而也会成为网络安全攻击的目标。一旦 DNS 服务器内容被篡改、连接被阻塞、系统被毁坏,大批用户的网络访问无法顺利进行,就会出现一种 Internet“假死”现象:用户习惯性地使用域名但无法访问网站,而实际上直接输入 IP 地址是可以访问的,问题是几乎没有人会把网站的 IP 地址记录下来以备不时之需。

计算机操作系统可为本机提供域名解析功能,方法是将 IP 地址和域名对应关系记录在 hosts 文件(为可编辑的文本文件)中,优先生效。但 hosts 文件有两个缺陷:IP 地址的改变无法自动获知与更新;可能被某些恶意程序利用,添加特定的解析条目,将网络访问引向陷阱网站的地址。例如,用户正确输入了某个网络银行的域名,但并不知道解析得到的 IP 地址是错误的,从而毫无戒备地在假冒网站上操作银行卡账号,后果不堪设想。

在访问网络资源时,域名只能粗略地指出访问目标的目的地计算机。为了更精确、有效地标识 Internet 上的各种资源,如网站、文件、设备等,Web 等应用引入了**统一资源定位符**(Uniform Resource Locator,URL),用于指示 Internet 上任何计算机的任何路径下任何格式的文件,包括虚拟化表示为文件的计算机外设等资源。URL 的语法格式如下:

应用协议://[账号:密码@][主机名.]域名或 IP 地址[:端口号][/路径][/文件名][?参数=值#标志]

URL 字符串一般是区分字母大小写的,也可由服务端设定为不加区分。

应用协议有 http、ftp、telnet、mailto、news、file 等,指明访问所用的应用层协议;账号和密码用于登录,不常用;主机名可省略,常见的有 www、mail 等;如果使用指定协议的注册端口号,则端口号字段可以缺省,否则就应指出所需的目的端口号;路径和文件名(或设备名)表述文件存放的详细位置,如果文件存放在根目录下,则可以省去路径,如果使用标准浏览器访问 Web,还可以省略文件名 index.html;参数字段用于查询数据库、搜索引擎等应用。

URL 即为 Web 所用的超链接(hyper-link)。例如:

```
http://www.fudan.edu.cn
http://baike.baidu.com/view/1496.htm
http://www.baidu.com/s?rsv_bp=1&wd=%D3%F2%C3%FB&inputT=2153
ftp://alice:112233@myfile.istore.net:8100/video/films/2012.mp4
https://pbnj.ebank.cmbchina.com/userLogin/Login.aspx
```

3. NAT

在 Intranet 内部可以使用 Internet 私有 IP 地址，为每一台计算机分配 IP 地址，而不一定要申请已经非常稀缺的公网（全网）IP 地址。依据这种策略可以有效地使用现有 IPv4 的地址空间，而且在很大程度上保障了内部网络的安全，因为从外部网络无法看到内部网络的情况，除非得到技术支持和许可，从外部网络无法访问内部网络设备，而内部网络设备则能够很容易地访问 Internet。

然而，使用私有 IP 地址是无法在 Internet 上进行寻址的。要让 IP 报文能够在 Internet 通行无阻，必须采用公网 IP 地址。所以，必须采用某种方法，当内部网络计算机需要和外部 Internet 进行通信时，转换为合法 IP 地址。实现这种 Intranet 地址规划及地址转换的方法就是根据 RFC 1631 设计的**网络地址翻译**（Network Address Translator，NAT）技术，用以完成内部私有 IP 地址与外部公网 IP 地址的代换工作。NAT 可以是单独的设备，但通常是集成在路由器中，成为路由器功能之一。

NAT 技术可以使用少量的（一个或多个）公网 IP 地址为 Intranet 上大量的计算机服务。值得关注的是，在实际情况下，从内部网络发起访问外部网络计算机的需求大大多于反过来发起的访问需求，因此，NAT 的设计是不对称的。

如图 3.8 所示，当内部网络的计算机 192.168.0.123 需要访问 Internet 上的服务器 166.125.77.231 时，计算机发送的 IP 报文可以顺利地发给服务器，但当服务器进行发送响应报文时，由于目的 IP 地址为私有地址，在 Internet 上是无效的，协议交互因此无法完成。所以，当 IP 报文从内网转发到 Internet 之前，应由 NAT 将私有地址翻译为公网地址（此例中为路由器的外部端口的注册 IP 地址 202.120.224.96）。可见，NAT **的出站翻译针对的是 IP 报文的源 IP 地址**。当返回的 IP 报文到达路由器后，再由 NAT 将报文中的目的 IP 地址恢复为原来的私有地址。

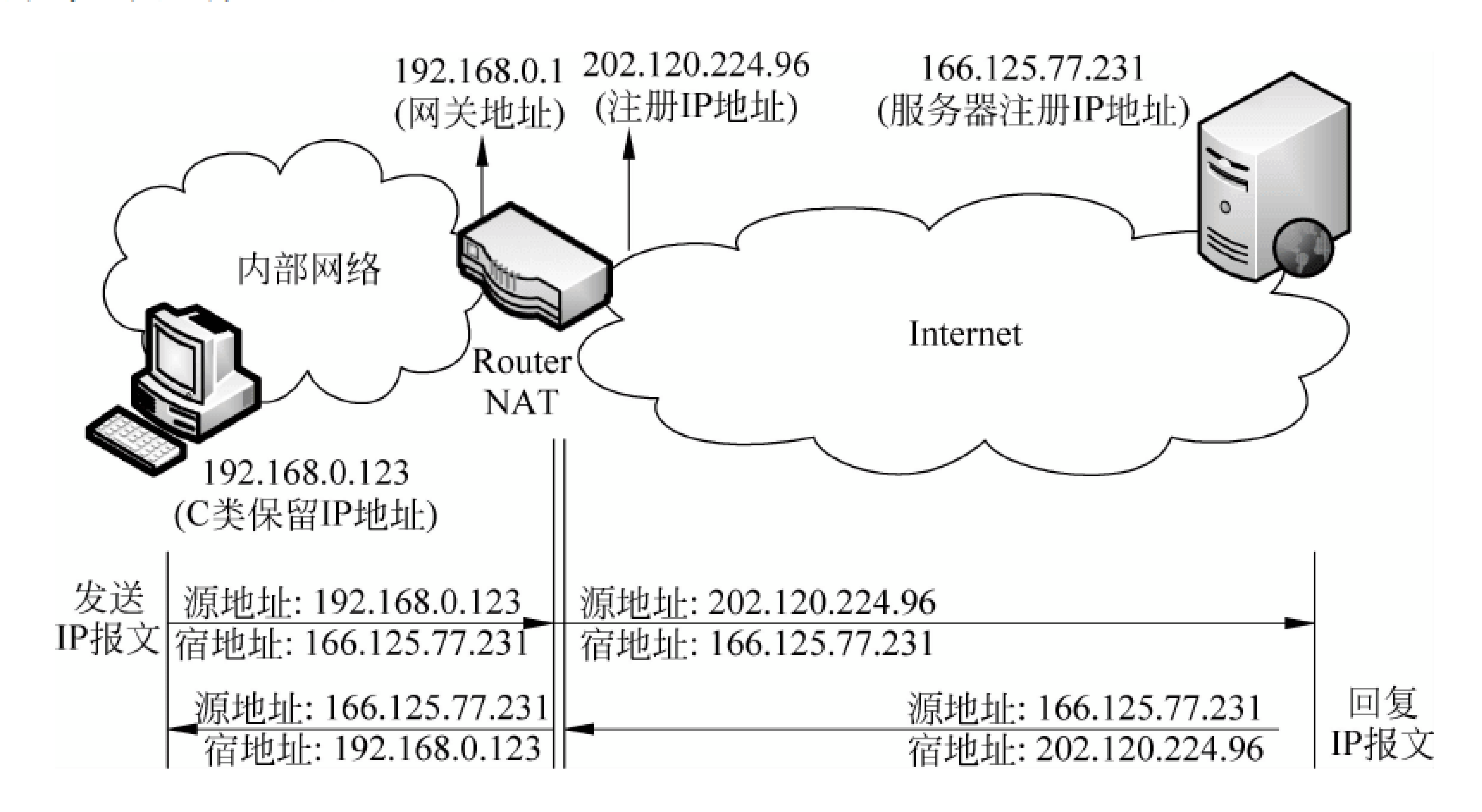


图 3.8　NAT 工作原理示意

NAT 有以下三种可选的技术方法。

（1）**静态翻译**（Static Translation）。静态翻译是指将需要外联的计算机的 IP 私有地址和公网地址一对一地翻译。有多少台内网计算机，就需要准备多少个公网 IP 地址。显而易见，这种方法的使用局限性很强，合法 IP 地址的利用率很低，只在少数特殊场合下应用。

思考:为实现 NAT 静态翻译,NAT 应维护何种用于运行管理的数据结构?

(2) **动态翻译**(Dynamic Translation)。当一台内部网络的计算机发起对外部网络的访问时,NAT 从公网 IP 地址池(address pool)中取得一个地址,由该访问连接使用,使用完毕后将该 IP 地址归还给地址池。在这种情况下,公网 IP 地址需要的数量可少于内网计算机的数量,比静态翻译的地址利用率高。然而,动态翻译方法看似简单,却存在许多不确定因素,实用性也不强。

思考:如何确定 IP 地址已经被使用完毕?

思考:当内部网同一台计算机访问不同的外部网计算机时,是否需要给予其不同的合法 IP 地址?

是否可能仅用一个公网 IP 地址,满足所有内网计算机的各种访问需求呢?

(3) **端口复用**(Port-Multiplexing)。不管是静态翻译还是动态翻译,IP 地址的利用和转换策略都显得比较粗线条,仅与 IP 地址信息相关联。互联网资源访问对应了不同的应用层协议,而各个应用层协议具有特定的 TCP/UDP 端口(port)。如果把端口信息及其映射机制结合到 NAT 技术中,就可以利用更多信息为算法服务,大大提高地址翻译方法的细粒度。

端口复用 NAT 方法的基本原理是:改变从内网到外网的 IP 报文的源端口号,替换为唯一编号,并将任意的私有 IP 地址、任意的网络应用都映射到同一个外部地址。这一算法又称为**端口地址翻译**(Port Address Translator,PAT)。

如图 3.9 所示,内网的两台计算机需要访问外网,三种 IP 报文体现了网络访问的两种典型需求:不同的计算机同时访问 Internet,同一台计算机同时进行访问 Internet 的不同应用。端口复用方法可以实现将三种 IP 报文翻译到同一个 IP 地址,并可以在报文返回时还原。

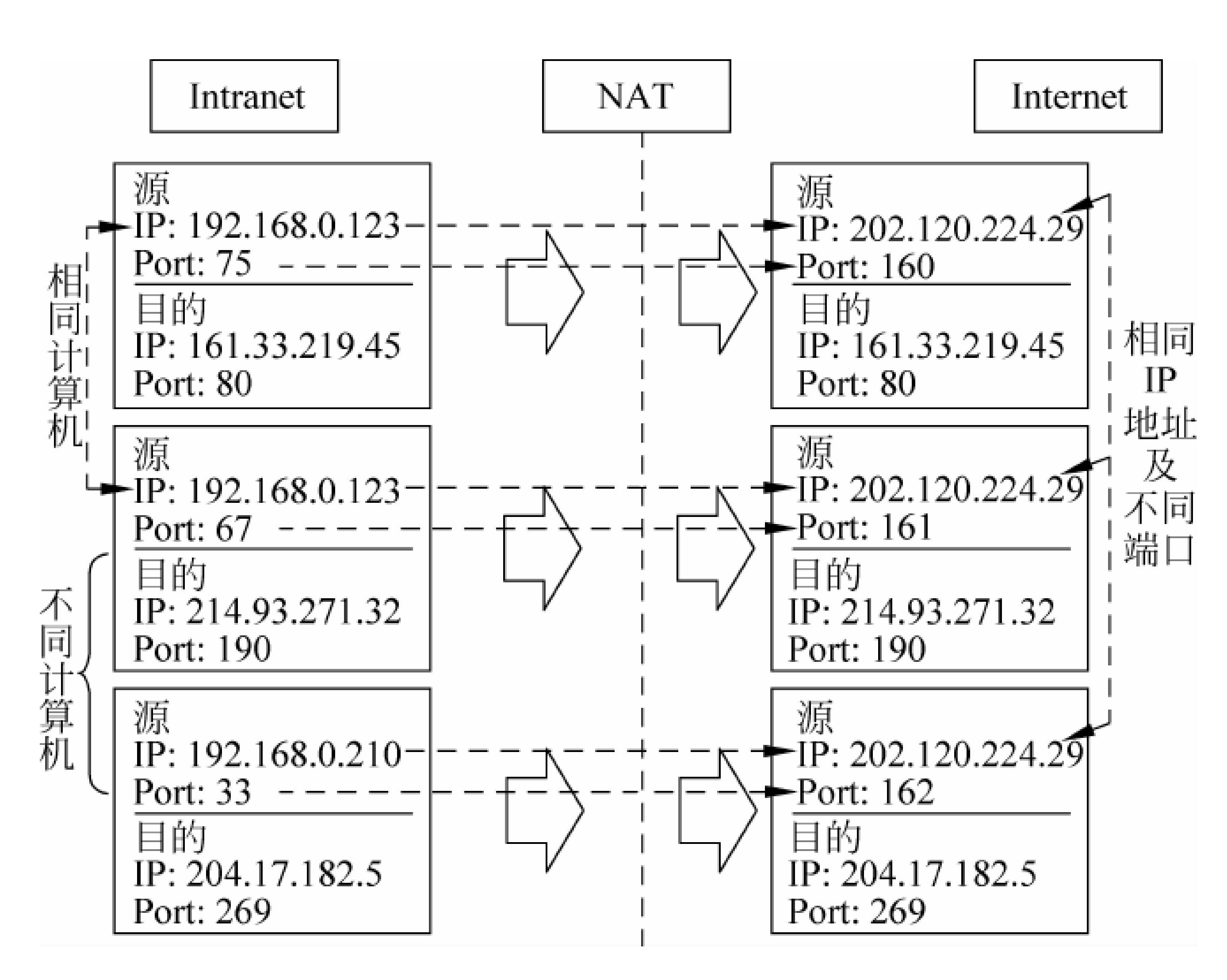


图 3.9　端口地址翻译原理

考察端口复用方法,凡是 IP 报文的源 IP 地址和源端口号相同的 IP 报文,端口号都被赋予同一个编号,只要源 IP 地址或源端口号有一者不同,其端口号就被赋予一个新的唯一

编号。NAT 将记录下这些转换关系，当 IP 报文返回时，就可根据目的端口号来找到出站时的转换关系，从而恢复目的 IP 地址和目的端口号。因此，出站 IP 报文的源端口号字段实际上是被 NAT 用以存放索引号，索引号可以随 IP 报文行走四方，一旦回家，就可以用唯一的索引号检索记录下的原始信息。

因为 TCP/UDP 的端口号为 16b，所以最多可以建立 65 536 个不同的索引关系，绝大多数情况下可以满足内部网络所有访问外网需求的不同源 IP 地址和源端口号组合的需要。即使发生编码不足情况，也可以淘汰长时间不活跃的组合。

3.2.2 IPv6

新一代 IP 的研究和提出源于对现有 IPv4 不足之处的认识。最主要的原因是 IPv4 地址几乎已经枯竭，无法应对 Internet 应用和用户数量迅猛增长的需要，对满足未来的增长预期更是无限悲观；其次，骨干网络上庞大的路由表拖慢了数据交换的速度，维护工作量十分惊人，而且越来越像一颗颗炸弹，随时可能致 Internet 于死地；此外，IPv4 对 QoS 缺乏支撑、安全性很弱、扩展选项过少、不支持即插即用等，都是颇受开发者和使用者诟病之处。

然而，IP 的技术更新并非只是一个纯技术问题，制约新一代 IP 全面替代 IPv4 的因素很多，最大的障碍恰恰是 Internet 本身。由于 Internet 非常成熟和稳固，又无与伦比地庞大，现有的绝大部分 IPv4 网络设备不可能轻易升级，也出于投资保护的考虑不可能全部更新。不论是作为 Internet 基石的内容提供者（ICP），还是作为 Internet 生命力的网络用户，绝大多数的服务器和个人计算机都是工作在 IPv4 上。虽然新版操作系统普遍支持新一代 IP 协议栈，但不可能一夜之间同时修改配置，结果就是 IP 升级被深层冻结、无限拖延。

IPv4 还在使用的主要原因是，采用了各种有效技术，如 ISP 对上网用户动态分配临时的 IP 地址、内部网络用户使用私有 IP 地址并用 NAT 访问外网等，提高了宝贵的公网 IP 地址的利用率。

1. IPv6 报文

IPv6 是 IP 的版本 6，又称为**下一代 IP**（IP next generation，IPng），由 RFC 2460～2463 定义。IPv6 是新一代网络的核心协议，新一代互联网（NGI）、新一代电信网（NGN）的发展可能是 IPv6 最终完全取代 IPv4 的机会。

如图 3.10 所示，IPv6 报文由基本首部（base header）、可选的扩展首部（extension header）以及承载数据所组成，但扩展首部不属于报文的首部（报头），而是与承载数据一起作为载荷（payload）来看待。由于路由器一般不处理扩展首部，因此，IPv6 既有较强的扩展性和灵活性、可提供更多的功能，又能保障网络结点处理的高效率。

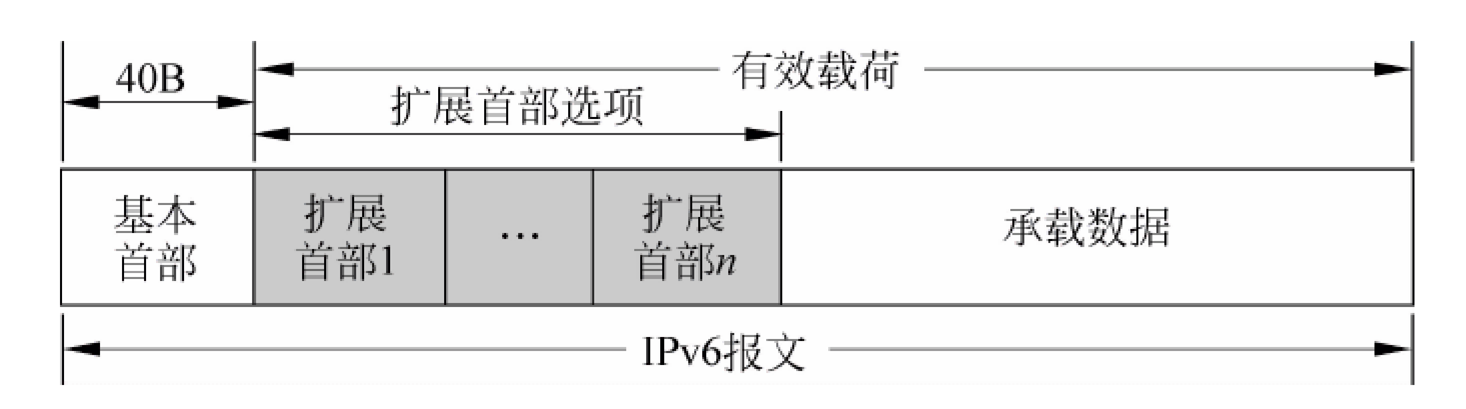


图 3.10 IPv6 报文一般格式

IPv6 的基本首部为固定 40B,分 8 个字段(如图 3.11 所示)。相比 IPv4 报头,取消了部分字段:有些已无存在必要(如报头长度),有些可以通过其他协议弥补(如校验和),有些则采用新的实现方法(如分片偏移量)。

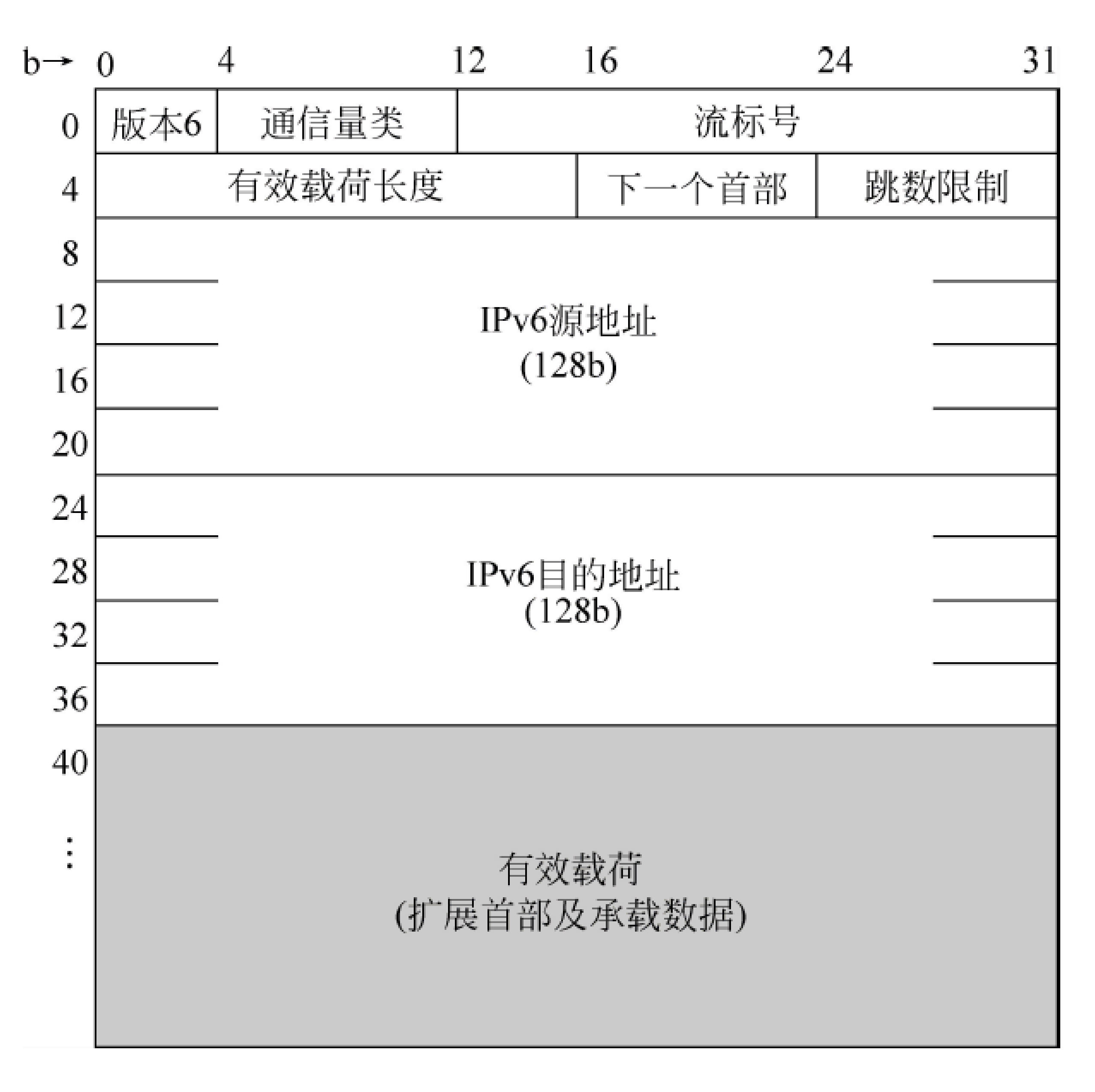


图 3.11 IPv6 基本首部格式

版本字段(version)为 4b,总是等于 6,表示版本 6。有效载荷长度为 16b,以 B 为单位,指示除基本首部外的其他部分的长度,最大值为 64K。跳数限制(hop limit)字段为 8b,与 IPv4 中的 TTL 相同,每经过一次路由器转发,其值减 1,直到归零时 IP 报文被丢弃,防止报文在网络中无休止地流转下去。

通信量类(Traffic Class)和流标号(Flow Label)分别占 8b 和 20b,是用于支持 QoS 机制和流量工程的参数。通信量类字段可区分不同的 IPv6 报文的类别或优先级,数值 0～8 表示阻塞控制业务量,当阻塞时可允许丢弃,数值 8～15 为非阻塞控制业务量,不期望被丢弃。流标号字段指明从特定源点到特定终点(单播或组播)的一系列报文(源和目的 IP 地址相同),尤其是实时信息传送,属于同一个流(flow)的报文有相同的流标号,流所流经的路由器保障其具有指明的 QoS。流标记可为任意伪随机数,不归属于任何流的报文中流标记置 0。

IPv6 源地址和目的地址各占 128b,即 16B,是 IPv4 地址长度的 4 倍、地址空间的 2^{96} 倍。IPv6 地址占有报头超过 3/4 的空间,是 IP 改进的重要目的和内容之一。

下一个首部字段(next header)为 8b,当有扩展首部时,该字段的值就是第一个扩展首部的类型,没有扩展首部时,该字段和 IPv4 的协议字段一样,指明承载数据的上层协议类型(如表 3.4 所示)。

IPv6 协议共使用 6 种扩展首部:①逐跳选项;②源站路由选择;③报文分片;④鉴别;⑤封装安全有效载荷(数据加密);⑥目的站选项。每个扩展首部均由若干字段组成,长度各不相同,但所有扩展首部的第一个字段都是 8b 的下一个首部(NH)字段。若有多个扩展首部,按照以上先后顺序排列。

表 3.4　IPv6 下一个首部字段编码

编码	协　　议	编码	协　　议
0	① 逐跳选项	43	② 源站路由选择
1	ICMP	44	③ 报文分片
2	IGMP	45	IDRP
4	IPv4	46	RSVP
6	TCP	50	⑤ ESP(加密)
7	UCL	51	④ AH(鉴别)
8	EGP	58	ICMP v6
9	IGP	59	无后续首部
17	UDP	60	⑥ 目的站选项

如图 3.12 所示，IPv6 报文允许无扩展首部和有扩展首部两种情况。

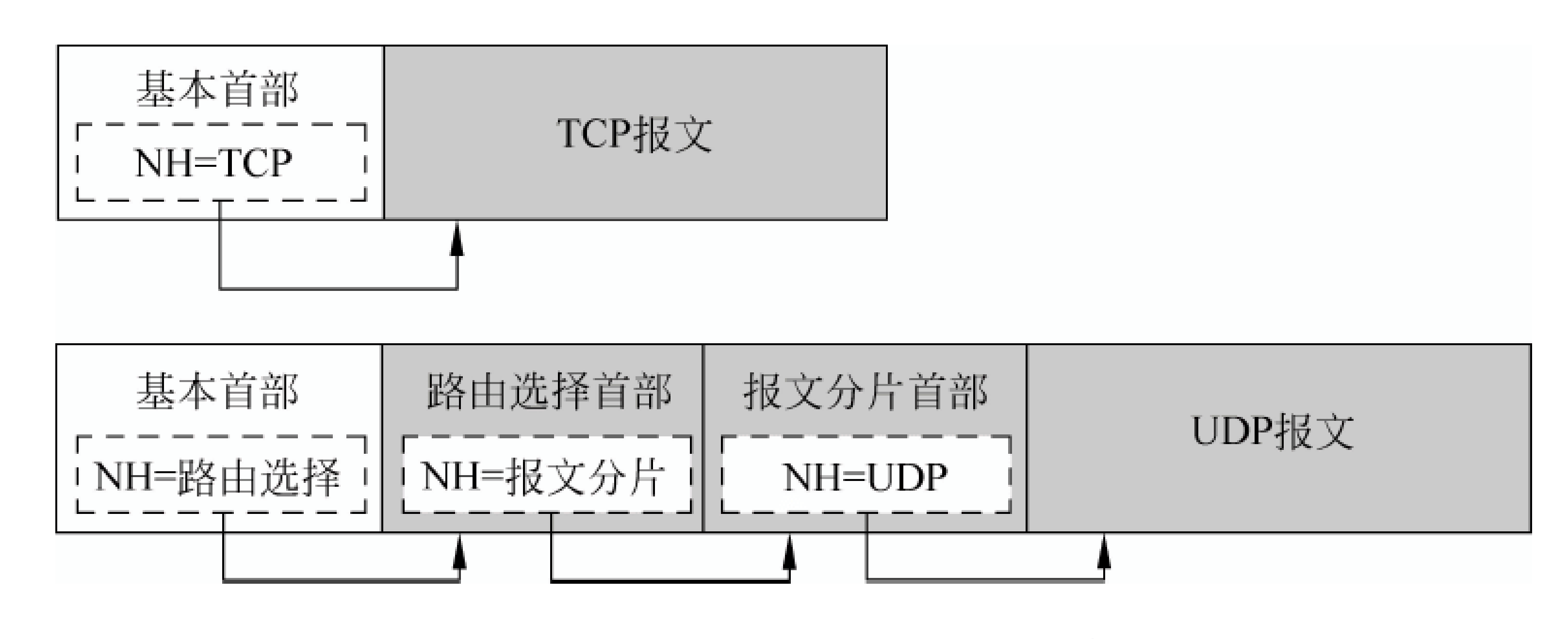


图 3.12　IPv6 报文扩展首部示例

IPv4 中，报文分片由报文分片扩展首部完成。报文分片一律由源发送站进行，是端到端的控制，路由器一般不参与分片。分片长度可根据数据链路层 MTU（Maximum Transmission Unit，最大传输单元）确定，或使用路径 MTU 发现（Path MTU Discovery）协议来确定。有些扩展首部不允许被分片，例如需要中间路由器处理的扩展首部。

如图 3.13 所示，报文分片扩展首部总共为 8B，其中分片偏移量字段为 13b，指出本分片在原来报文中的偏移量，以 8B 为计量单位，这说明每个分片的长度一定是 8 的整数倍。M 字段是 1b 的更多分片（more）标志，$M=1$ 表示后面还有分片，$M=0$ 表示为最后一个分片。

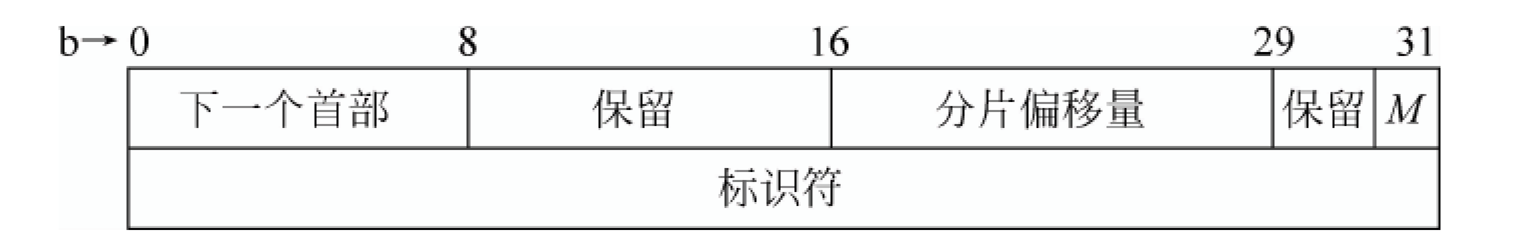


图 3.13　IPv6 分片扩展首部格式

标识符字段是 32b 的数值，由源发送站生成，一般每产生一个新的原始 IP 报文，标识符加 1，保证发送到相同目的站点的报文在数据报的生存时间内标识符具有唯一性。一个原始报文的所有分片报文继承该标识符，即归属相同的原始 IP 报文的分片的标识符是相同的。

设：承载 TCP 的原始 IP 报文的有效载荷长度为 LB，当前标识符计数器为 C，需要分成长度为 NB 的若干个分片，应有 $L=kN+n$，其中 nB 为最后一个分片的长度。

又设：在基本首部中，以 len 表示有效载荷长度字段，以 nh 表示下一个首部字段；在扩展首部中，以 NH 表示下一个首部字段，SO 表示分片偏移量字段，ID 表示标识符字段，则分片方式如图 3.14 所示。

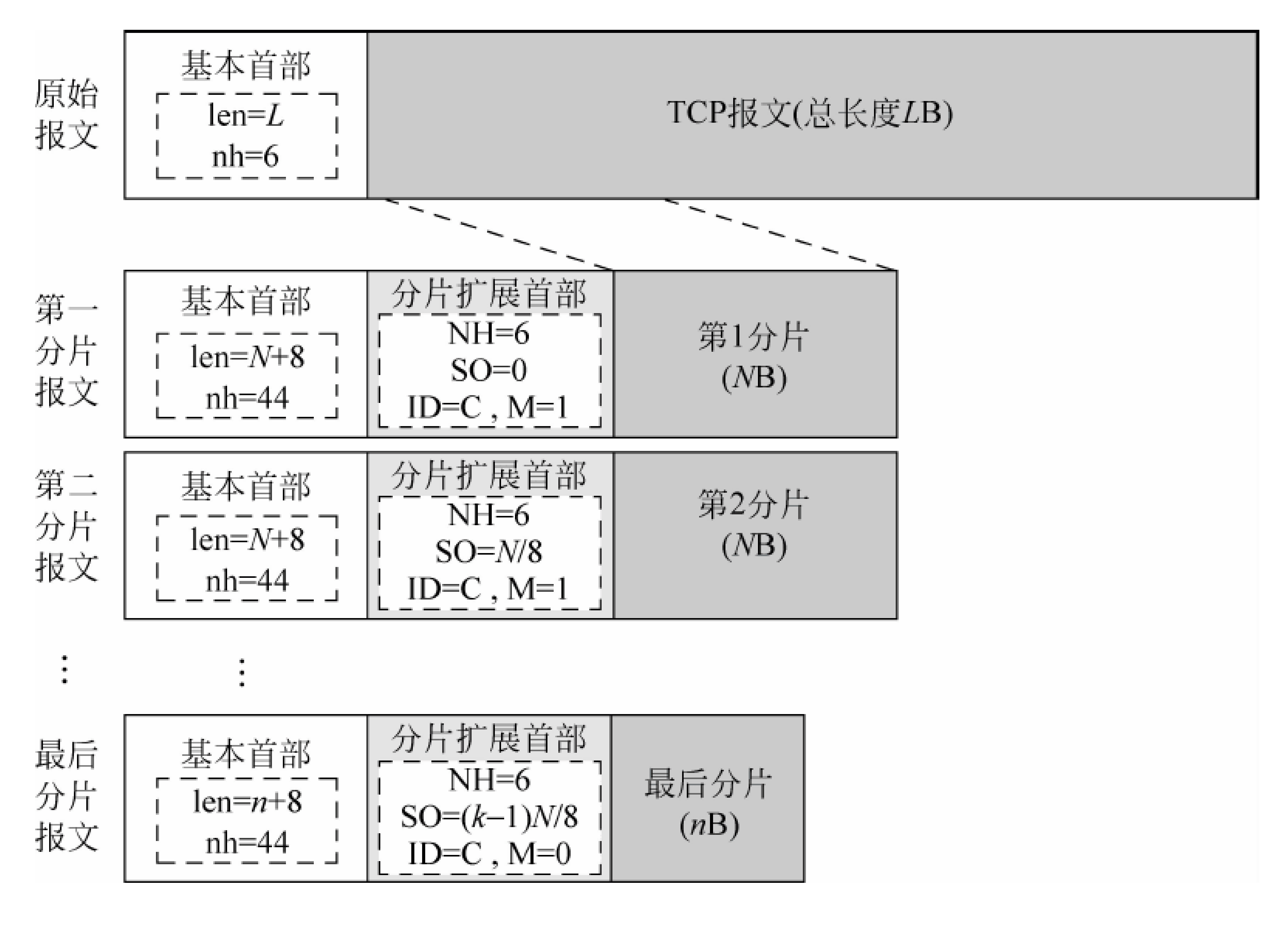


图 3.14　IPv6 报文分片示例

如果由于路由变化，IPv6 报文大于新的发送链路的 MTU，路由器可以采用隧道分片法，连同基本首部在内一起参与分片，即进行二次分片(封装式分片)，在链路另一端由 IPv6 协议机进行组装，恢复原分片。

源站路由选择功能用于指定转发路径，其扩展首部格式如图 3.15 所示。其中，地址字段长度以 8B 为计算单位；路由选择类型定义为 0；路段偏移指示取值 0～23，指向当前路段的对应的 IP 地址，即源站指定的路径所应当经过的 1～24 个路由器，每经过一个路由器，路段偏移指示加 1；映射比特字段的每一个比特对应于 24 个地址中的一个，若某个比特为 1，表示是严格的源站选路，即该比特所对应的地址必须成为其前一个地址的下一站地址，反之，若某个比特为 0，表示是松散的源站选路，即该比特所对应的地址不一定必须是其前一个地址的下一站地址。

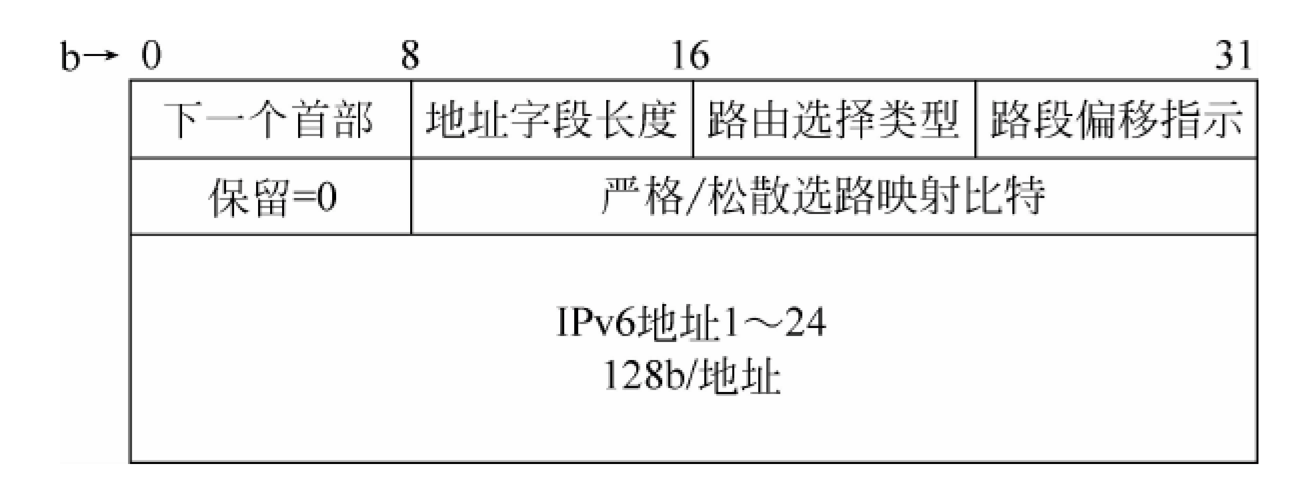


图 3.15　IPv6 源站路由选择扩展首部格式

另外，逐跳选项扩展首部用于传送由报文的传送路径中各个结点检测的可选信息；目的站选项扩展首部用于携带只需由报文的目的结点检测的可选信息；鉴别扩展首部 AH 由

RFC 1825 和 RFC 1826 规定；数据加密扩展首部 ESP 由 RFC 1825 和 RFC 1827 规定。

2. IPv6 地址

与 IPv4 的 32b(2^{32}＝4 294 967 296)地址空间相比，IPv6 拥有达 128b 地址空间。

$$2^{128}=340\ 282\ 366\ 920\ 938\ 463\ 374\ 607\ 431\ 768\ 211\ 456$$

这个数量相当于地球上每平方米有 7×10^{23} 个地址；如果地址使用频度是 100 万个/μs，需要 10^{19} 年才能用完。这足以使 IP 地址分配远离捉襟见肘的尴尬境地，可以保证每个人的每个随身电器都可以获得一个公网 IP 地址。

当然，IPv6 地址空间的扩大是有代价的，IP 报头从 20B 增加了一倍到 40B，因此，每个 IP 报文都要耗费更多带宽。

IPv6 地址用冒号十六进制记法，如：

3457:76C8:F:5432:813:0:FFFF:7359

地址中间部分连续的 0 可以作零压缩，如 7A33::E9、C3D:54::。任意地址零压缩只能用一次，防止出现错误理解。冒号十六进制记法也可与 IPv4 地址的点分十进制记法混合使用，用于 IPv4 和 IPv6 的转换中，如 0:0:0:0:0:0:192.168.18.1，零压缩为::192.168.18.1。

在表示 IPv6 超网时，CIDR 地址前缀的记法也可进行零压缩，例如 60b 前缀的 IPv6 地址，以下三种记法等价：

35B8:0000:0000:4A90:0000:0000:0000:0000/60 或

35B8::4A90:0:0:0:0/60 或

35B8:0:0:4A90::/60

URL 中 IPv6 地址可表示为 http://[1088::5:503A:450:87C9]:80/index.html。

IPv6 地址有三种类型，除了与 IPv4 相同的**单播**(unicast)、**组播**(multicast)外，新增加了**任播**(anycast)类型。任播的目的站是一组计算机，但报文只交付给其中的一个站点，如距离最近的一个。IPv6 将**广播**(broadcast)看做组播的一个特例。

一个结点(主机、路由器)可以拥有多个 IPv6 单播地址。

IPv6 地址分为两个部分：可变长度的类型前缀定义了地址的目的，其余部分长度也随之可变。RFC 2373 定义的类型前缀如表 3.5 所示。

表 3.5　IPv6 地址类型前缀定义

类型前缀	地址类型	所占份额
0000 0000	保留(与 IPv4 兼容)	1/256
0000 0001	—	1/256
0000 001	保留给 NSAP 地址	1/128
0000 010	保留给 IPX 地址	1/128
0000 011	—	1/128
0000 1	—	1/32
0001	—	1/16
001	可聚合的全球单播地址	1/8
010	基于网络供应商的单播地址	1/8
011	—	1/8
100	基于地理位置的单播地址	1/8

续表

类 型 前 缀	地 址 类 型	所 占 份 额
101	—	1/8
110	—	1/8
1110	—	1/16
1111 0	—	1/32
1111 10	—	1/64
1111 110	—	1/128
1111 1110 0	—	1/512
1111 1110 10	本地链路单播地址	1/1024
1111 1110 11	本地站点单播地址	1/1024
1111 1111	组播地址	1/256

IPv6 地址中前缀 0000 0000 用于与 IPv4 地址兼容和映射。兼容地址用于自动隧道技术中,使用这种地址的结点既支持 IPv4 也支持 IPv6;映射地址用于不支持 IPv6 的结点。地址转换方法如图 3.16 所示。

比特数→	80	16	32
IPv4兼容的IPv6地址	0000 ··· 0000	0000	IPv4地址
IPv4映射的IPv6地址	0000 ··· 0000	FFFF	IPv4地址

图 3.16　IPv6 地址与 IPv4 地址兼容和映射方法

IPv6 地址具有 3 个等级(如图 3.17 所示):

第一级,顶级聚合,全球可寻址的地址(top level);

第二级,站点级聚合,本地站点(单个地点)地址(site level);

第三级,本地链路(网络接口)地址(link level)。

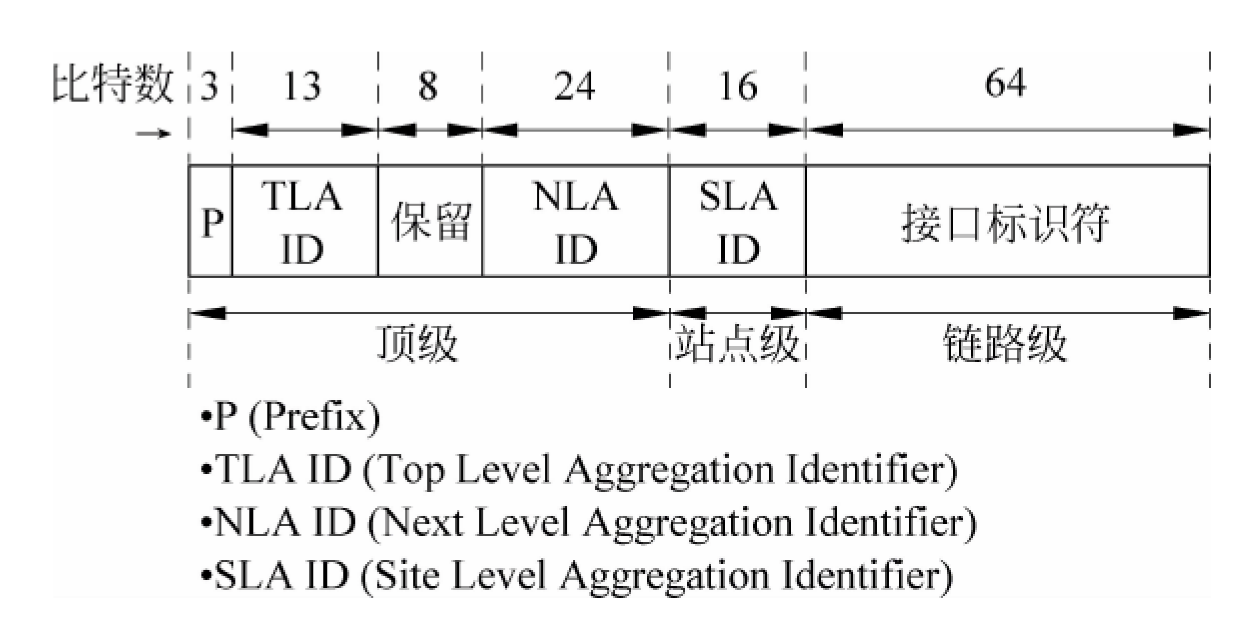


图 3.17　IPv6 单播地址等级结构

IEEE 的 64b 地址 EUI-64 可直接放入 IPv6 地址的接口标识符中完成转换,但要将 EUI-64 地址的第一字节最低第 2 位(G/L 比特)置 1。

IEEE 802 以太网的 MAC 地址为 48b,与 IPv6 地址转换方法如图 3.18 所示,其中 c 为企业标识符的比特,g 为 I/G 比特,G/L 比特为 0。

对于可聚合的全球单播地址,格式前缀 P=001,TLA 分配给 ISP 或拥有这些地址的汇聚点(exchange),NLA 分配给特定的用户(subscriber)。SLA 和 IPv4 中的子网类似,对应

于在一个地点的一组计算机和网络，相距较近，且由一个单位管理（相当于局域网或内部网）。

接口标识符即为计算机和网络的单个接口。IPv6 使用一种邻站发现协议（Neighbor Discovery Protocol）来确定相邻接的结点（ICMPv6 中包含），就是采用了硬件地址（如 MAC 地址）映射的方法，而不需要采用 ARP（Address Resolution Protocol，地址解析协议）解析。

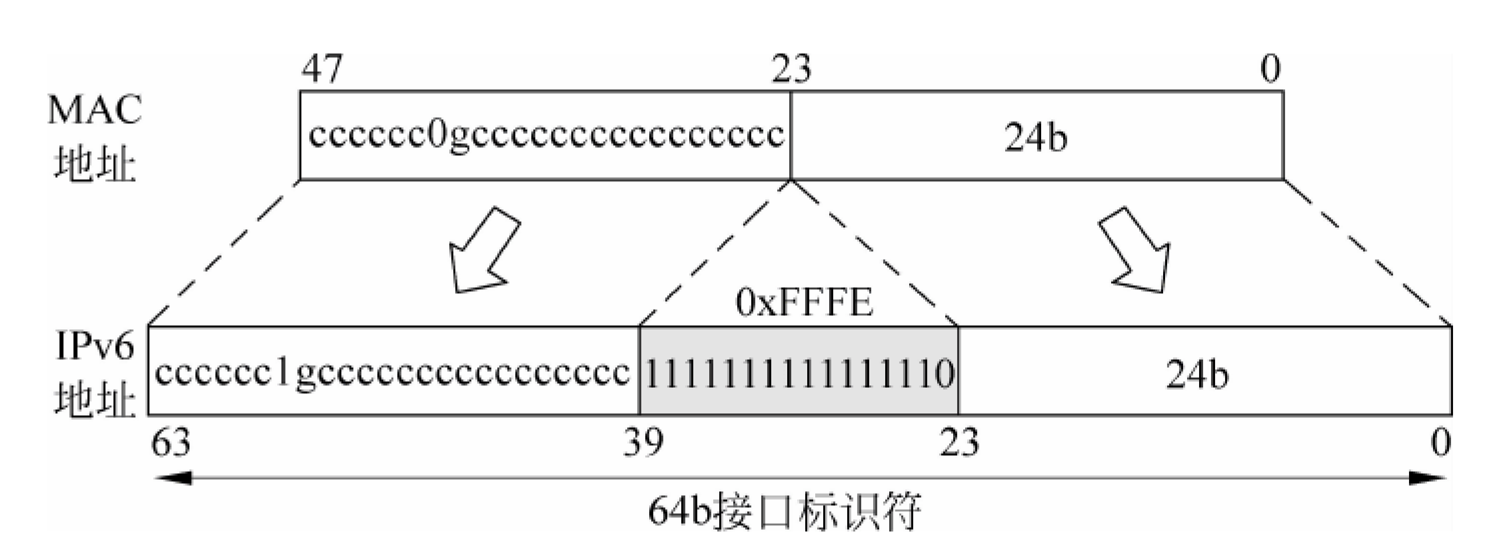


图 3.18 MAC 地址转换为 IPv6 地址

3.2.3 TCP/UDP

1. TCP

传输控制协议（Transport Control Protocol，TCP）是 OSI 模型的第四层运输层协议，由 RFC 793 规定。

TCP 和 IP 是非常理想的搭档，IP 提供寻址和路由，并能适应各种通信系统（数据链路层、物理层），而 TCP 则通过虚电路和流量控制等操作，很好地弥补了 IP 不可靠传输的缺陷。因此，TCP 成为许多 Internet 应用层协议的网络服务提供者。

相对 IP 而言，TCP 比较复杂。为完成建立连接、传输数据等功能，需要相应的 PDU 来传递控制信号。TCP 的 PDU 报头结构如图 3.19 所示，最少 20B，主要字段说明如下。

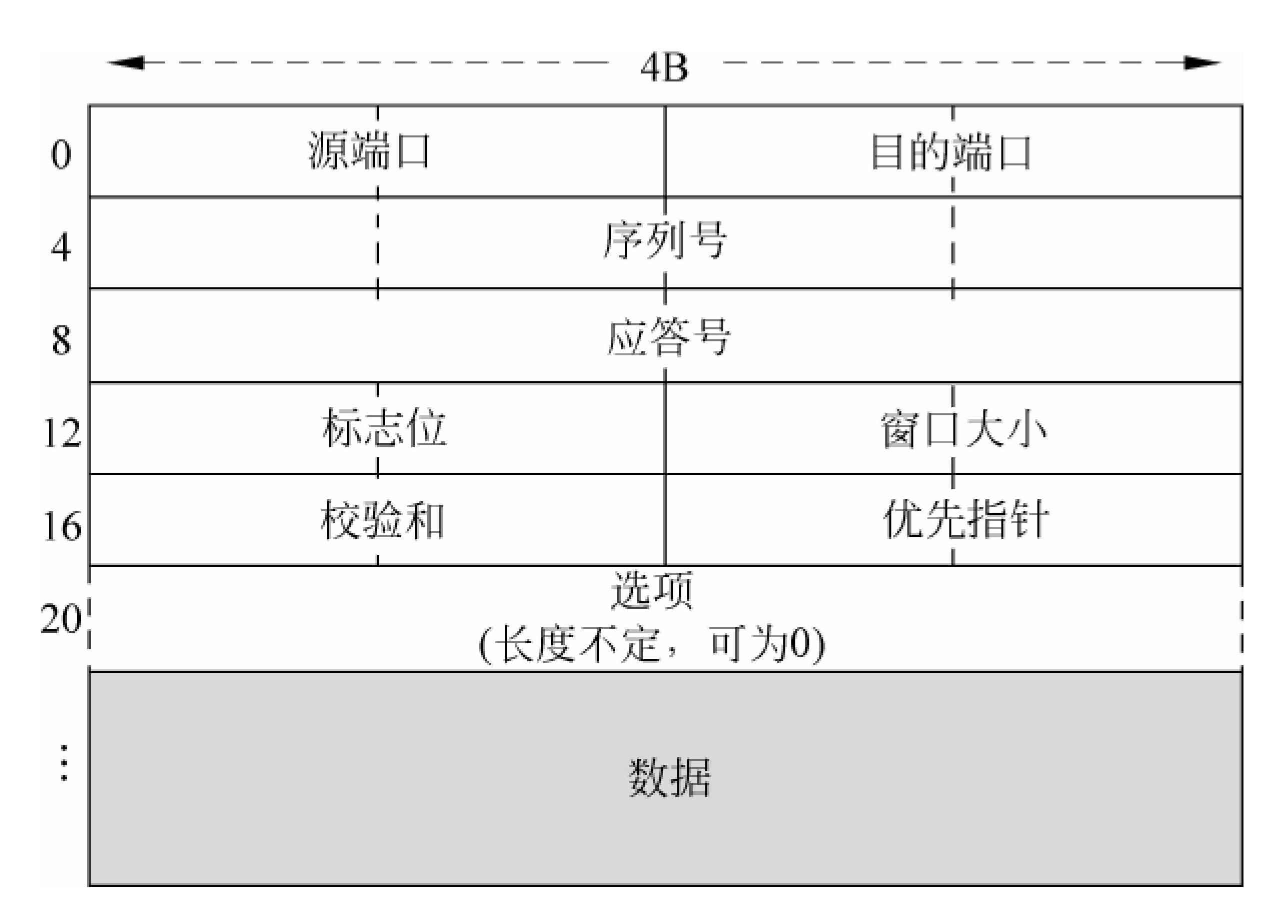


图 3.19 TCP 报文

（1）源端口（Source Port）：16 位的源端口号，其中包含初始化通信的端口。源端口和源 IP 地址的作用类似，用以标示报文的返回点。

（2）目的端口（Destination Port）：16 位的目的端口号定义传输的目的，指明接收报文

的计算机上的应用程序目标。

(3) 序列号(Sequence Number):32位的序列号由接收端计算机使用。当SYN(同步)开始时,序列号设定为初始序列码(Initial Sequence Number,ISN),而第一个数据字节是ISN+1。序列号可以补偿传输中的不确定情况。

(4) 应答号(Acknowledgment Number):32位的序列号由接收端计算机使用,如果设置了ACK控制位,该值表示下一个准备接收的序列号。

(5) 标志位(Code Bits):包括4位报头长度(HLEN)、6位保留位和6位标志位,具体为紧急标志URG、应答标志ACK、推送标志PSH、复位标志RST、同步标志SYN、完成标志FIN。

(6) 窗口大小(Window Size):16位,表示已准备好接收的每个TCP数据段的最大长度。

(7) 优先指针(Urgent Pointer):16位,指向后面是优先数据的字节,在URG标志设置时才有效。

鉴于TCP的复杂性,仅讨论连接过程。一个TCP连接被称为一次会话(session)。建立TCP连接就是创建会话的过程。由于TCP连接需要经过"请求→响应→确认"三个阶段,因此称为三次握手(triple-handshaking)。简要连接建立过程如图3.20所示。

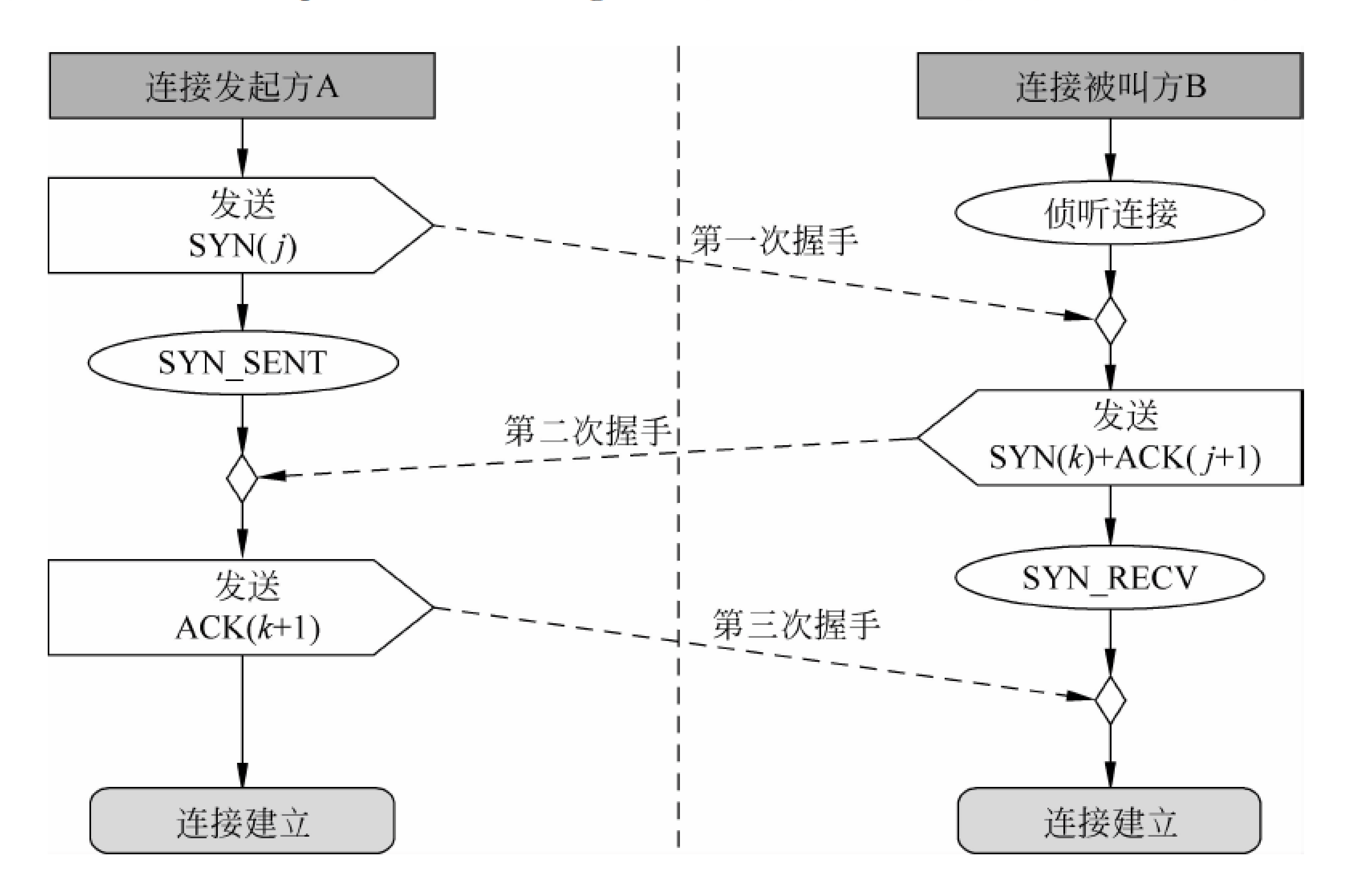


图3.20 TCP连接三次握手示意

第一次握手:发起方A发送SYN报文(syn=j),A进入SYN_SENT状态,等待对方B响应。

第二次握手:B收到SYN报文,必须确认A的SYN(ack=$j+1$),同时自己也发送一个SYN包(syn=k),即SYN+ACK,B进入SYN_RECV状态,等待A确认。

第三次握手:A收到SYN+ACK报文,向B发送确认包ACK(ack=$k+1$),A和B均进入ESTABLISHED(连接建立)状态,完成三次握手。

2. UDP

用户数据报协议(User Datagram Protocol,UDP)由RFC 768定义,一方面和TCP一

样，属于 OSI 的第四层运输层，另一方面，就如其名 Datagram，又和 IP 一样，是面向非连接的数据报传输协议。

如图 3.21 所示，UDP 的报文结构非常简单，长度为 8B 的报头仅有最为必要的 4 个字段，而且其中只有与 TCP 一致的源端口和目的端口提供了运输层应有的服务功能。



图 3.21 UDP 报文

相比 TCP，UDP 几乎是“零协议”，那么，为什么拥有功能完备的 TCP 的同时，还需要 UDP 呢？如表 3.6 所示，两个协议的差异很明显，各有特点，在一定程度上回答了这个疑问。

表 3.6 TCP 和 UDP 比较

	TCP	UDP
协议复杂性	较复杂	很简单
逻辑连接	必须先建立连接	不必建立连接
数据传输	可靠	不保证可靠
流量(拥塞)控制	支持	不支持
占用计算资源	较多	很少
协议机状态	有多种状态	无状态
对网络服务要求	较高质量	无要求
对计算机性能要求	较高性能	较低性能
网关穿透能力	较弱	较强
传输实时性	较差(时延较大)	较好(时延较小)
支持应用层协议	FTP、HTTP、Telnet、SMTP 等	TFTP、SNMP、IPSec、RIP 等

思考：环境温度测量系统应采用 TCP 还是 UDP 协议？为什么？

3. Socket

TCP/UDP 使用**端口**(port)来区分应用层协议、识别不同的进程。例如，IP 协议机向上层递交报文，如果发现端口号为 21，则将交给 FTP 进程。所以，端口号相当于进入各自归属地的通行证号码。IETF IANA 定义了三种端口组(常见的端口号分配如表 3.7 所示)。

(1) 公认端口(Well-known Ports)：0～1023。

(2) 注册端口(Registered Ports)：1024～49 151。

(3) 动态/私有端口(Dynamic/Private Ports)：49 152～65 535。

在 Internet 应用系统中，客户端计算机通常不关心端口号。作为应用活动的发起方，客户端程序可以自行分配动态或私有的源端口号(Source Port)，只需保证该端口号是本计算机上唯一的，然后可侦听该端口号来获取返回的报文。而规定的端口号(公认端口或注册端口)一般用于服务器端，作为报文中的目的端口号(Destination Port)，以体现用户真正的访问意图。

表 3.7　常用 TCP/UDP 端口号

端口号	TCP/UDP	应用层协议	中文名称与简要说明
7	TCP/UDP	Echo	回显
19	UDP	CharGen	返回随机字符报文
20/21	TCP	FTP	文件传输/控制
23	TCP	Telnet	远程登录
25	TCP	SMTP	电子邮件
53	TCP/UDP	Domain	域名服务
67/68	UDP	DHCP	DHCP(Bootstrap)服务器/客户端
69	UDP	TFTP	简单文件传输
70	TCP	Gopher	Gopher
79	TCP	Finger	Finger
80	TCP	HTTP	WWW 超文本传输
110	TCP	POP3	电子邮件"邮局"(版本 3)
123	UDP	NTP	网络对时
143	TCP	IMAP4	消息访问
161/162	UDP	SNMP	网络管理/告警(Trap)
179	TCP	BGP	边际网关路由
194	TCP	IRC	聊天室
389	TCP	LDAP	轻量目录访问
443	TCP/UDP	SHTTP	安全 HTTP
500	UDP	IPSec	密钥交换
513	TCP	rlogin	远程登录(UNIX)
520	UDP	RIP	路由协议 RIP(版本 1 和 2)
531	TCP	Conference	会议聊天

思考：如果发出报文的端口号为$[P_s, P_r]$，那么接收到的回复报文的端口号是怎样的？

在网络通信过程中，仅仅依靠 TCP/UDP 端口号是不够的。端口号必须与目的 IP 地址相结合才有实际意义。因此，应用需求实际上包含了两个要素：网络地址和应用类型。前者指示"去哪里"，后者说明"干什么"。

TCP/IP 协议栈一般为应用层协议提供一种网络服务接口，使应用程序可以根据需求进行调用，获取特定的数据传输服务、发送和接收相关的报文。

如图 3.22 所示，TCP/IP 网络服务接口就是**套接字**(Socket)。

Socket 是 IP 地址和 TCP/UDP 端口号的结合体，可以描述为地址矢量{IPaddr，port}，IPaddr 决定目标计算机，port 决定应用及其进程。Socket 也可描述为一个四元组[srcIPaddr，srcPort，dstIPaddr，dstPort]。在网络上，具有相同四元组的 IP 报文属于同一个流(flow)。

Socket 是一套接口规范，包括以下技术要点。

(1) Socket 可看做 TCP/IP 协议机上的一层外壳(shell)，可以屏蔽许多无关的协议细节，提供友好的、规范的网络服务。

(2) Socket 可向应用层协议提供创建和关闭连接(对于 TCP)、全双工传输数据等功能。

(3) 接收方可侦听 Socket 接口,接收指定端口的报文。

(4) Socket 具备两种数据服务类型(实际上就是两种运输层协议)。

① 流式数据 Sock_Stream: 提供连续的、可靠的、双向的、基于连接的字节流数据传输机制,使用 TCP。

② 块式数据 Sock_Datagram: 提供面向非连接的、不保证可靠的、成块的数据传输机制,使用 UDP。

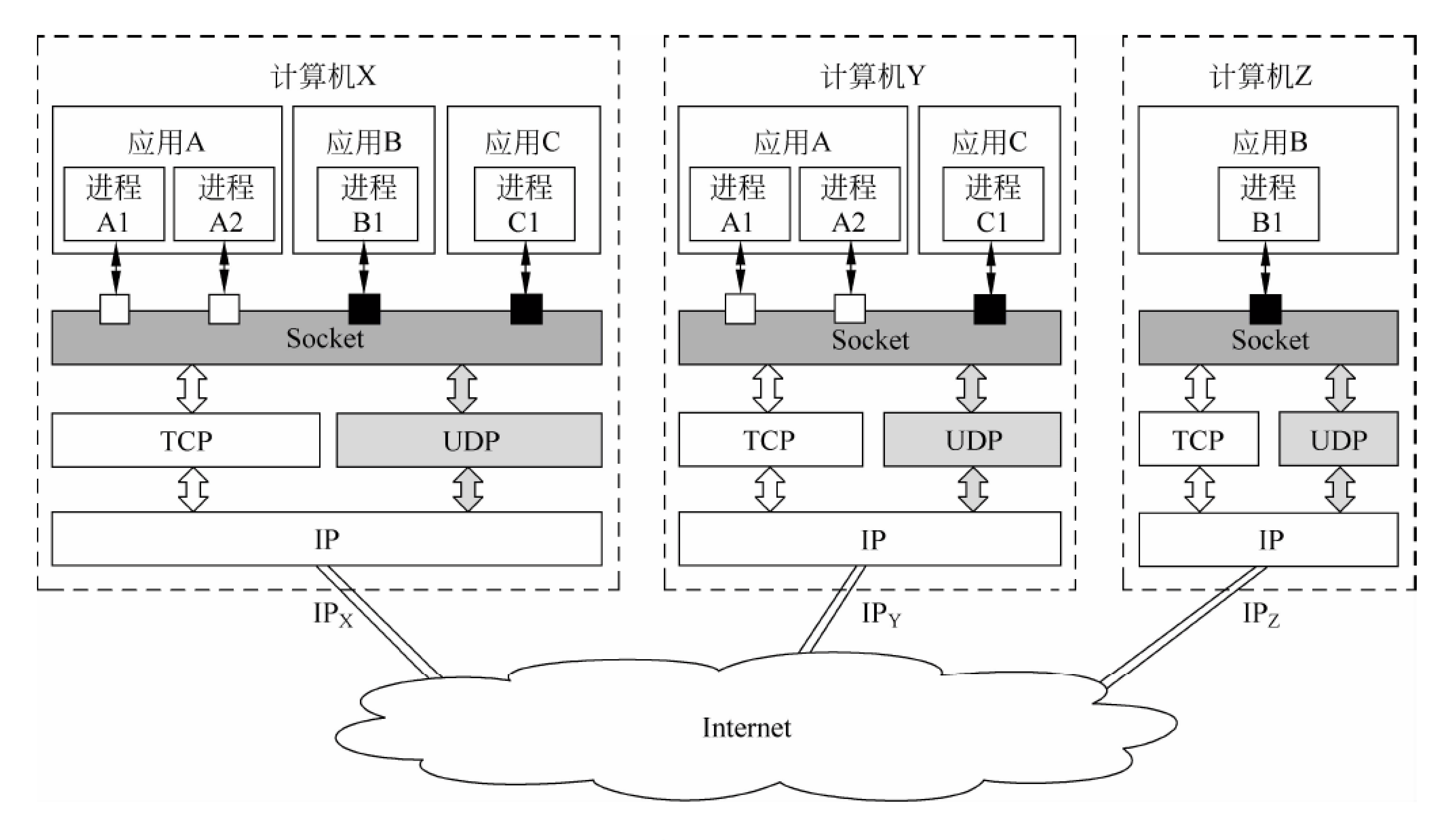


图 3.22　套接字接口服务原理

3.3　Internet 典型应用协议

3.3.1　Telnet

远程登录协议(Telnet)用于通过 Internet 远程访问并操作计算机主机,如进行远程登录操作系统、远程执行计算任务等。Telnet 协议标准由 RFC 854/855 规定,是最早期的 Internet 应用层协议,至今仍然在发挥重要作用。

计算机的多用户、多任务操作系统允许多个(可达数十个)用户终端设备(terminal)连接并进行交互式运行。对于每一个用户而言,仿佛独占着计算机,实际是处在操作系统管理下的相对独立的虚拟用户空间(虚拟主机)中。多用户接入方式经历了三个阶段的发展。

(1) 简单终端。终端由显示屏和键盘组成,通过异步串行通信接口(com)与主机连接,只能进行命令行操作。受通信距离的局限,终端必须部署在主机周围数十米的范围内。

(2) 仿真终端。使用 PC 模拟简单终端的输入输出,仍然采用 com 接口,但可支持窗口式用户操作界面,提高了访问便捷性。

(3) 远程终端。计算机终端(本质上是客户端程序)通过 Internet 与主机互连,采用 Telnet 协议,支持命令行或窗口式远程操作,还可采用后台隐藏式自动运行方式。

Internet 的 E-mail 系统最初就是采用 Telent 登录到服务器，输入并运行操作系统命令来完成电子邮件的编辑、发送、接收、处理等任务。

Telnet 协议为客户机/服务器互连模式，运行在 TCP 上(端口号为 23)。Telnet 客户机与服务器是通过 TCP 传送命令(字符串)和返回响应信息的方式实现交互。客户机具有两种状态：命令方式和会话方式。前者让用户对 Telnet 进程(本地客户机)发布指令，完成建立连接、释放连接等操作；后者情况下用户输入的命令被交付给远程计算机处理。系统运行时，两种状态可使用相应的命令进行转换。

Telnet 客户机始终是会话过程的发起方和控制方。初始情况下，客户机处于命令方式，运行 open+<地址或域名>命令发起 TCP 连接，要求接入 Telnet 服务器；网络连接成功后，自动进入会话方式；远程服务器首先回送 login 和 password 字符串，提示用户传送正确的用户名(账号)和口令(密码)；如果账号验证正确，用户就可进一步发送操作命令，远程控制服务器的运行；所有操作任务完成后，输入 close 命令结束会话(释放 TCP 连接)，回到命令状态；输入 quit 命令将完全退出 Telnet 应用程序。

Telnet 程序可设置为字符传送模式(每个输入字符一个 TCP 报文)或字串传送模式(多个字符或整条命令合并传输)。服务器可提供字符回显(echo)功能，兼容简单终端的操作。

思考：*计算机有哪些方法能够阻止 Telnet 访问？*

3.3.2 FTP

文件传输协议(File Transfer Protocol，FTP)用于实现各类文件的可靠传输，由 RFC 959 标准定义，运行在 TCP 上(端口号 20/21)，是最早的应用层协议之一。由于计算机的各种内容，如文档、音乐、图片、电影、软件等均以文件形式存储，获取内容就是获得文件，因此 FTP 在 Internet 上的应用非常广泛和频繁。

FTP 采用客户机/服务器互连结构，并有主动(active/port)和被动(passive)两种通信方式。主动方式下，数据连接由服务器发起，被动方式下服务器被动监听端口，连接均由客户机发起，有利于穿越客户端的防火墙限制，例如大部分 Web 浏览器都要求使用被动 FTP 方式。

主动方式的 FTP 建立连接过程如下。

(1) FTP 客户端打开一个随机的 TCP 端口 x(x>1024)，作为命令端口，连接至服务器的端口 21，用于传输控制流。

(2) 客户端启动监听数据端口 $x+1$，同时用端口 21 向服务器报告正在监听的端口号并且已准备好从该端口接收数据。

(3) 服务器打开端口 20，发起与客户端数据端口 $x+1$ 的 TCP 连接(被动方式下服务器等待连接)。连接成功后用于传输数据流。

(4) 客户端接受连接建立，然后向服务器发送应答消息，告知连接已经准备好。

连接建立后，FTP 可选择字节流模式、数据块模式或压缩模式传输文件数据，数据可为 ASCII 编码、EBCDIC 编码或二进制编码(binary，例如图像、可执行文件)，支持非结构化文件、结构化记录和分页式数据。

FTP 可使用命令实现用户登录(user/password/account)、显示文件列表(list)、修改当前路径(cwd/cdup)、更改文件名(rnfr/rnto)、删除文件(dele)或退出(quit)等控制操作，当

然最核心的操作是实现文件传送。RETRIEVE(retr)用于客户端从 FTP 服务器获取文件，即常用的下载(download)；STORE (stor)用于从客户端传送文件给服务器，即上传(upload)。

许多公共的 FTP 服务器支持匿名登录方式，以 anonymous 为用户名，不需要口令，存在较大的安全隐患。与 Telnet 应用类似，FTP 也分为字符型命令行界面和窗口型图形化界面，后者可由专门的客户端软件实现(可便捷地单击或拖动文件)，也可由 Web 浏览器及其插件实现。

简单文件传输协议(Trivial File Transfer Protocol，TFTP)是一种简化的 FTP，于 1980 年定义为 RFC 1350 标准。TFTP 通常被较低性能的计算机用于传输小文件。TFTP 的部分技术基于更早期的 EFTP，而 EFTP 是通用分组协议(PUP Protocol)的一部分。

TFTP 使用 UDP(端口号 69)为传输协议，不提供用户登录、目录列表等功能。TFTP 协议通信过程如下。

(1) 客户机发送读请求(RRQ)或写请求(WRQ)给服务器，其中包含文件名和传输模式(ASCII 或二进制编码)。

(2) 服务器发送 ACK 应答，指出了数据报文所用的端口号。

(3) 发送方向接收方传送经编号的数据报文，除最后一个外，都为全尺寸的文件数据块，每次只发送一个报文(发送窗口为 1)。最后的数据报文必须包含少于全尺寸的数据块，以表明文件传输结束；如果被传输的文件正好是全尺寸块的整数倍，则最后应发送一个 0 字节数据块的报文。

(4) 接收方用相应编号的 ACK 应答每一个数据报文，即为半双工的乒乓式协议。

3.3.3 SMTP/POP

电子邮件(Electronic Mail，E-mail)从 1971 年诞生起，一直是 Internet 应用的常青树。

当用户注册了 E-mail 账号后，就拥有了全网唯一的电子邮件地址，格式为：

用户名@域名

E-mail 结构非常类似于传统的信件，分为信封(信头)和信体两个部分。信封包含了发信人地址、收信人(包括抄送或密送的收信人)地址、主题、优先级和日期等字段；信体包含了信件内容和附件。

@符号恐怕是 Internet 上最美的字符之一。它简洁、生动、直观，正好是 at(在……)的缩写，不论是书写、朗读还是计算机解析，都显得完美无瑕。电子邮件应用的设计者之一雷·汤姆林森(Ray Tomlinson)在谈到为何选择@符号时曾说过：“我只不过看了看它(@)，它就在那里(键盘上)。我甚至没有尝试其他字符。”好一次美丽的偶遇。

相比之下，URL 中的“://”组合符号则显得臃肿、丑陋和不知所云，完全是面向软件工程师的技术格式，而不是面向使用者的友好形式。不仅输入时费时费力，保存和传输时浪费了更多字节。

E-mail 采用 Internet 应用层的**简单邮件传送协议**(Simple Mail Transfer Protocol，SMTP)发送邮件。SMTP(RFC 821/822)使用 TCP(端口号 25)在 E-mail 服务器之间、电子邮局和 E-mail 服务器之间运行。

E-mail 系统没有传统意义上的用户端客户机,用户需要使用 Telnet 协议登录到其账号归属的 E-mail 服务器,然后使用相关命令来编辑和发送邮件、查看接收到的邮件。为了方便用户操作,专门设计的窗口式邮件客户端软件(如 Outlook)起到电子邮局的作用,采用 RFC 1081/1939 **电子邮局协议**(Post Office Protocol version 3,POP3),工作在 TCP(端口号 110)上,实现自动获取 E-mail 服务器上收件箱中的邮件并存放在计算机中供随时处理,而发送邮件仍然采用 SMTP。另一种常用方法是采用浏览器访问 Web Mail,以 Web 服务器作为邮件收发代理,用户通过操作网页来接收和发送电子邮件。

Internet 邮件访问协议(Internet Message Access Protocol version 4,IMAP4)是 POP3 的改进协议,提供了邮件检索和处理的新功能,用户不必下载邮件正文就可以看到邮件的标题、摘要,可以使用客户端软件对服务器上的邮件和文件夹等进行操作。

如图 3.23 所示,E-mail 服务器系统包括三个关键模块。

(1) 邮件传输代理(Mail Transport Agent,MTA):负责把邮件由一个服务器传送到另一个服务器的邮件投递代理。

(2) 邮件投递代理(Mail Delivery Agent,MDA):把接收到的邮件分发到用户的邮箱中。

(3) 邮件用户代理(Mail User Agent,MUA):协助用户读写邮件。

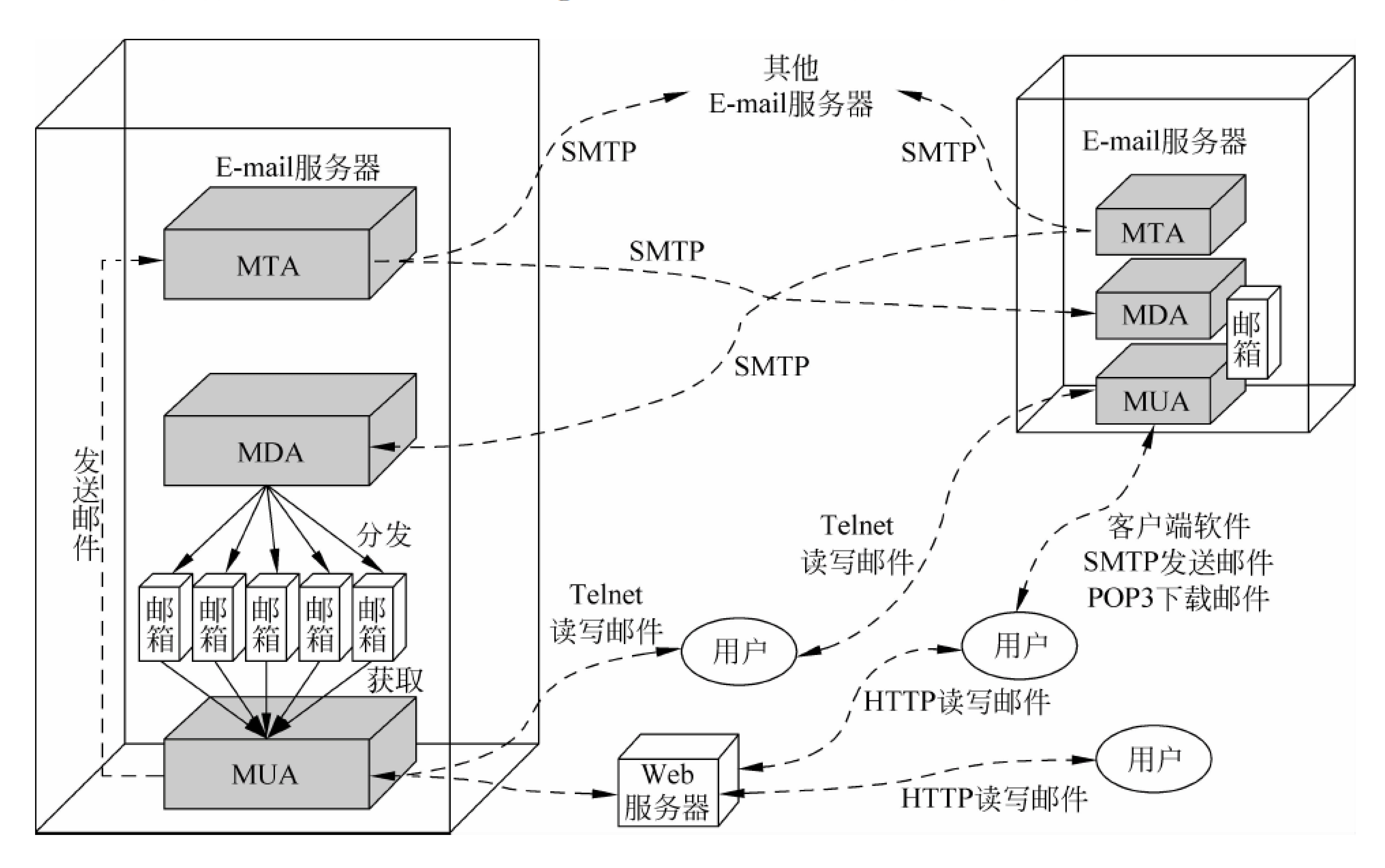


图 3.23　E-mail 系统组成结构示意

除了基本的邮件收发功能外,E-mail 系统可支持邮件列表(mail list)、自动回复、定时发送、外部邮箱 POP 代理等扩展功能。

E-mail 技术经过多年的发展,逐步从单纯的文本型邮件演化出超文本邮件、多媒体邮件、安全邮件(可加密和电子签名)。在移动计算领域,出现了面向手机用户的 WAP Mail 和 Push Mail(实现邮件主动推送功能)。

由于 Internet 上 E-mail 通常为免费、匿名的服务,邮件发送成本低、效率高,因此,常被

人用来群发广告信、欺诈信等垃圾邮件(spam)。据统计,垃圾邮件占 E-mail 总量的 80%~90%,成为 E-mail 系统挥之不去的噩梦。

思考:试比较邮件客户端软件和 Web Mail 在应用上的优势和劣势。

3.3.4 HTTP

万维网(World Wide Web,WWW)简称 Web,虽然在 Internet 应用中出现较晚(1990 年),但发展迅猛。Web 体现了瘦客户机的思想,而且具备跨平台的优良技术特性,已经成为最主要的信息发布和访问手段。

如图 3.24 所示,Web 可以视为架设在 Internet 的一张虚拟的网络,网络的结点为 Web 网站(Web site)及其保存的 Web 网页(Web page),网络的连接为超链接(hyper-link),即指示各类资源位置的 URL。各种信息相互交叉关联,形成网状结构,将 Internet 上的资源联成一体,便于搜索和访问。

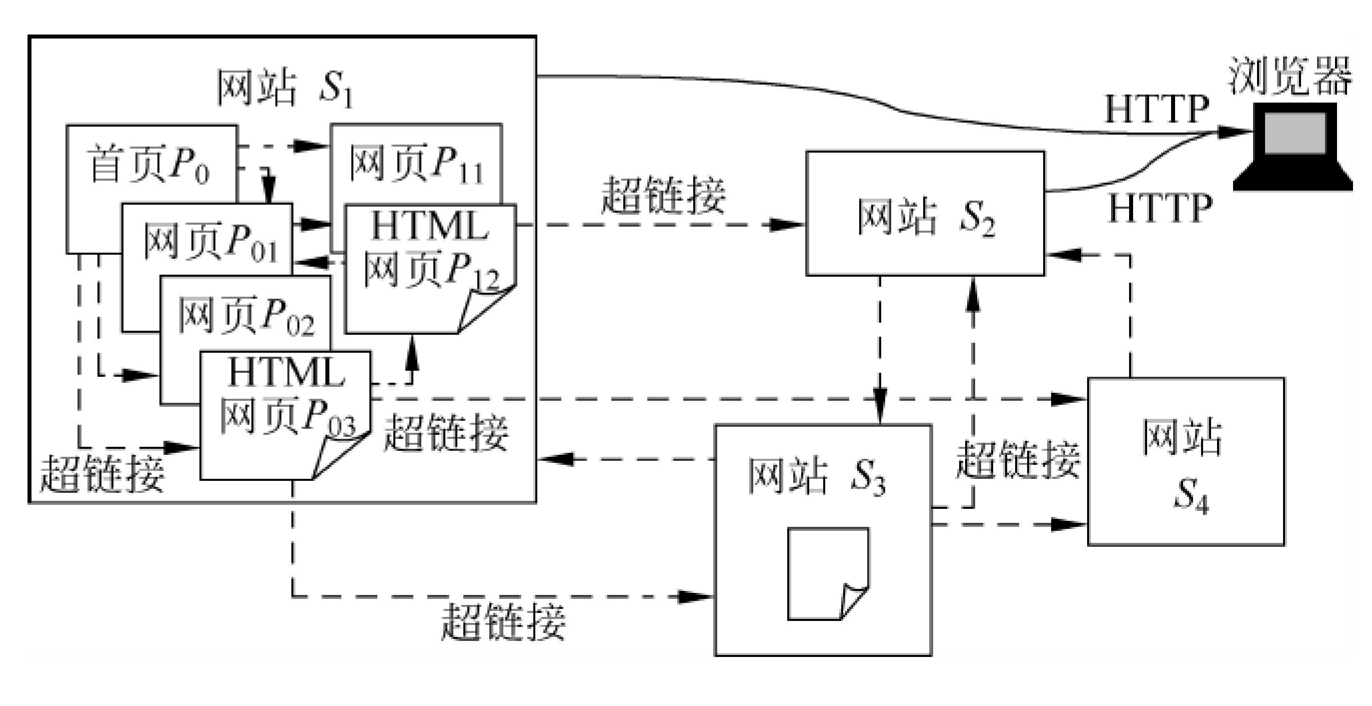


图 3.24　Web 网络原理

Web 网页遵循 RFC 1866 定义的**超文本标记语言**(Hyper-Text Markup Language,HTML)标准,为结构化的文本文件,使用不同的标记来表达各种多媒体信息、显示格式和访问操作。如下所示为一个简单的 Web 网页。

```
<html>
<head>
<title>欢迎访问</title>
</head>
<body>
<center> <p> <a href = "http://www.fudan.edu.cn">复旦大学</a> </p> </center>
<center> <p> <img src = "fudan_gate.jpg"> </p> </center>
</body>
</html>
```

通常一个网站的首页(或称主页)文件为 index.html,存放在根目录下,其他网页文件和所需的图片、音频等文件存放在各级目录下。

Web 访问使用标准化的浏览器(browser)。Web 浏览器可以解释每一张网页包含内容的含义,并按网页设计要求显示在浏览器窗口中。浏览器本质上是一个虚拟机软件,在不同

操作系统环境下 HTML 网页有相同的显示效果,而且逐步扩展到可以运行嵌入在网页中的脚本程序和 Java 程序。

浏览器负责访问用户指定的网站,并采用 RFC 1945 定义的**超文本传输协议**(HyperText Transfer Protocol,HTTP)来下载网页文件。HTTP 运行在 TCP 上,默认端口号为 80,具有如下特点。

(1) HTTP 属于面向对象的协议,适用于分布式超文本信息访问。浏览器作为客户机只需向 Web 服务器传送请求方法和路径,协议简单,降低了程序复杂度,执行速度快、效率高。

(2) HTTP 允许传输由 content-type 标记的任意类型的数据对象,灵活性强,可满足各种应用需求。

(3) HTTP 是无状态协议,不需要记忆事务处理进程的每个环节,不需要进行复杂的同步协作;每次 TCP 连接只处理一个请求,服务器处理完请求并收到客户机的应答后即断开连接。

HTTP 主要分为客户机(浏览器)发送的请求报文和服务器发送的响应报文。

请求报文由三部分组成:请求行(request line)、请求头(request header)、请求体(request body)。如果报文有请求体,则在请求头和请求体间插入一个空行。请求报文为文本型,每行都以 CR-LF(回车和换行符)结尾。

请求行格式为:

```
<method> <URL> <version>
```

其中:method 为请求方法,包括 get(获取资源)、post(附加新数据)、head(获取响应报头)、put(存储资源)、delete(删除资源)等;version 为 HTTP 版本号,例如 get /Depts/cse. html HTTP/1. 1。

思考:*HTTP 请求行的 URL 中为何不需要域名?*

请求头用于服务器需要的附加信息;请求体可为所需的任意数据。两者均可由多行字符串组成。例如一个 post 请求报文如下所示:

```
post /login. jst HTTP/
Accept: image/gif, image/jpeg
Accept - Language: zh - cn
Host: www. fudan. edu. cn
Content - Length: 21
Connection: Keep - Alive
Cache - Control: no - cache

user = alice&pwd = 112233
```

Web 服务器接收到请求报文后,将返回 HTTP 响应报文。响应报文也由三部分组成:状态行、响应头、响应体(响应体前插空行)。

状态行格式为

```
<version> <status - code> <reason - phrase>
```

其中：version 为 HTTP 版本号；status-code 为响应的状态代码；reason-phrase 为状态代码的文本描述。例如：HTTP/1.1 200 OK。状态代码 1xx 表示请求已接受，继续处理；3xx 表示需要进行重定向。其他常见的状态代码及其描述有：

- 200 OK(请求成功)；
- 400 Bad Request(请求有语法错误，无法理解)；
- 401 Unauthorized(请求未经授权)；
- 403 Forbidden(服务器收到请求，但拒绝提供服务)；
- 404 Not Found(请求的资源不存在，例如提供了错误的 URL)；
- 503 Server Unavailable(服务器无法访问)。

若 HTTP 请求成功，则网页等资源将在响应报文的响应体(实体)中传送。请求和响应报文都可以传送实体。一个实体由实体报头域和实体正文组成，但并不一定要一起发送，可以只发送实体报头域。实体报头定义了关于实体正文(如是否有实体正文)和资源的元信息。例如 Expires 指明过期日期和时间，让代理服务器或浏览器按时更新缓存。

从 HTTP 的角度看，Web 浏览器和服务器之间为非对称的客户机/服务器(client/server，C/S)结构，但 Web 信息访问的技术为信息系统的构造提供了不同于传统 C/S 方式的新思路、新架构，即浏览器/服务器(browser/server，B/S)方式。如图 3.25 所示，对任意应用系统，用户只需使用浏览器，通过 Web 服务器的代理来访问应用程序，形成三层架构(3-tier)，在用户操作便捷性和友好性、程序运行高效性、多系统集成、软件维护和升级、网络及数据安全等方面具有明显优势。

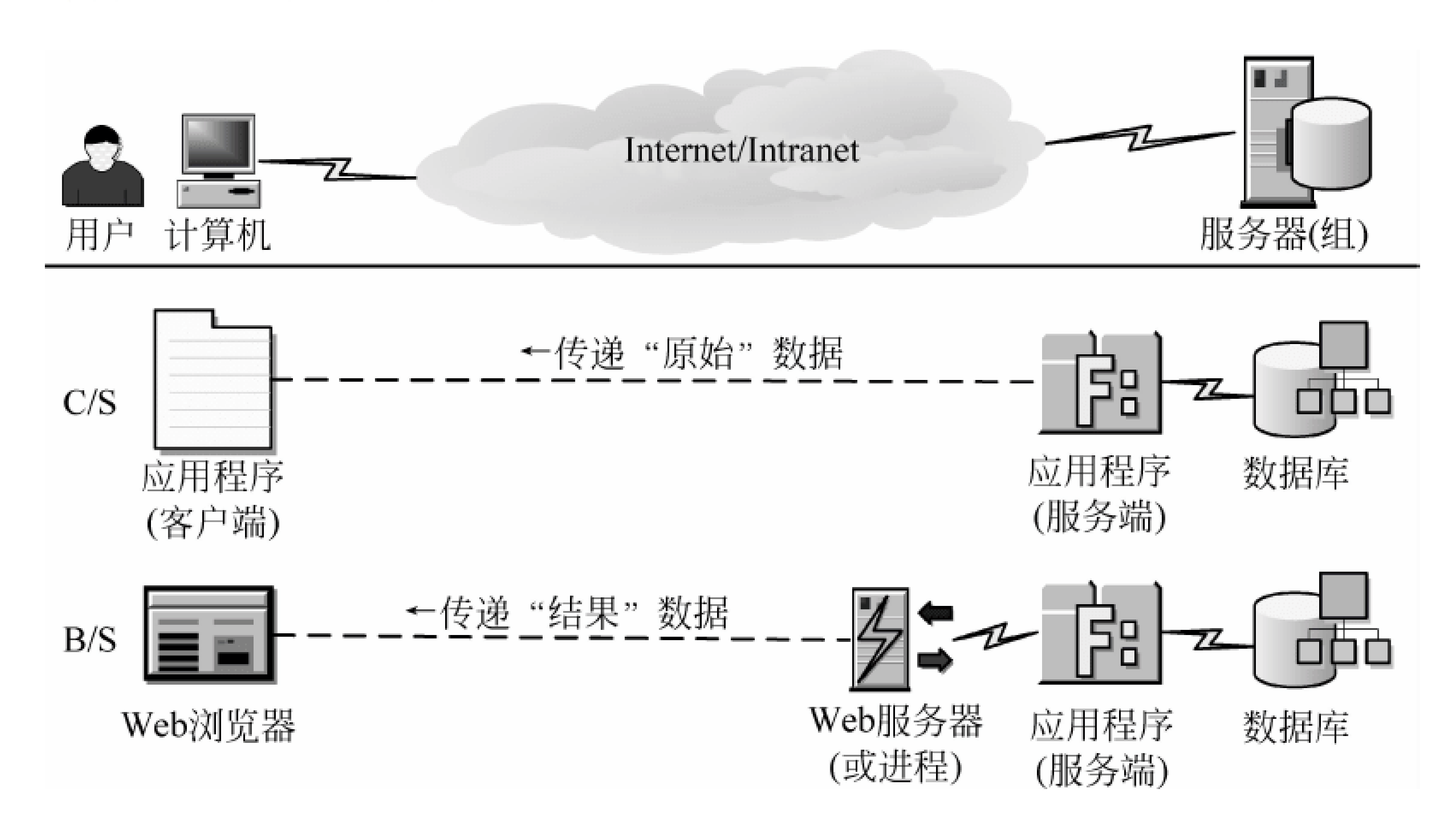


图 3.25　C/S 与 B/S 方式简要比较

思考：比较 C/S 和 B/S 方式的异同点，举例说明分别适合哪些应用。

3.4　Internet 控制和管理协议

3.4.1　ARP

一个典型的 IP 内部网络是由计算机(包括 PC 和服务器)、接入 Internet 的路由器所构

成的LAN,在最常见的联网方式下,计算机和路由器采用Ethernet集线器或二层交换机互连,位于同一个子网(网络号相同)。

假定计算机和路由器都已配置了内网IP地址,然而,Ethernet集线器或二层交换机无法识别IP,那么,设备如何将报文发送给接收方?

Ethernet网络采用MAC地址(俗称硬件地址)进行数据转发,所以,如果能够建立计算机IP地址与MAC地址的关联,那么根据IP地址就可以查找到对方的MAC地址,然后通过Ethernet网络送达目的设备。这一机制就是IP-MAC地址映射表,包含一系列{IP地址;MAC地址}记录。计算机和路由器都应该缓存本子网所有结点的地址映射,发送IP报文时即可随时检索。

可是,当有新成员加入网络时,其他成员并不了解其MAC地址,新成员也没有其他成员的MAC地址。为了建立或更新IP-MAC地址映射关系,就需要使用地址**解析协议**(Address Resolution Protocol,ARP)。ARP是Internet网络层的一个子层,在RFC 826标准中定义,用于辅助IP工作。IPv6中已不使用ARP,而以ICMPv6的相应功能取代。

在IP over Ethernet情况下,ARP的报文数据结构如下:

```
{
    short arp_hrd;          /* 16b 硬件类型,Ethernet */
    short arp_pro;          /* 16b 协议类型,IP 协议 */
    char arp_hln;           /* 8b 硬件地址长度,= 6B */
    char arp_pln;           /* 8b 协议地址长度,= 4B */
    short arp_op;           /* 16b ARP 操作类型,请求或响应 */
    char arp_sha[6];        /* 8×6b 发送者的硬件地址,源 MAC 地址 */
    long arp_spa;           /* 32b 发送者的协议地址,源 IP 地址 */
    char arp_tha[6];        /* 8×6b 目标的硬件地址,目的 MAC 地址 */
    long arp_tpa;           /* 32b 目标的协议地址,目的 IP 地址 */
}
```

ARP协议报文直接作为Ethernet网络的MAC帧数据,MAC帧的协议字段(16b)指示承载数据为ARP。ARP请求报文中未知的目标硬件地址字段不必赋值。

ARP的工作流程如下。

(1) 发送方计算机优先检索地址映射表,若找到与IP报文目的地址相同的IP地址,则使用该记录中的MAC地址进行发送。

(2) 若映射表中检索不到该目的地址δ,ARP将在Ethernet上广播发送查询报文,报文的意思是“我的IP地址为α,MAC地址是β,请问IP地址δ的MAC地址是什么?”

(3) 目的方计算机收到ARP查询报文后,立即向发送方回复响应报文(单播方式),报告自己的IP-MAC地址映射(顺便把询问方的地址映射记录下来)。其他计算机不作响应。

(4) 询问方收到该响应报文后,即可更新地址映射表,返回步骤(1)。

(5) 若在规定时间内得不到响应,可能是因为IP地址不正确,或目标设备不在线,或网络故障,或征询报文丢失,结果都将导致ARP查询失败。若尝试多次仍然失败,则终止IP报文发送。

每条映射记录可附加TTL(生存时间,可为15~20分钟),超时后记录失效,有利于减小映射表长度,提高检索效率,并有利于适应计算机更换的情况(例如IP地址用于新计算机)。

如果 IP 报文的目的地址不属于本子网(网络号不同),则须通过路由器进行转发。路由器的内网 IP 地址事先已经在计算机上配置好,但路由器 Ethernet 端口的 MAC 地址仍然需要通过 ARP 获得。需要特别注意的是,对于这些 IP 报文,并非根据其目的 IP 地址来检索地址映射表,而是使用路由器的 IP 地址来检索,然后使用路由器的 MAC 地址来封装和发送 IP 报文。

此外,早期的某些计算机设备(如无盘工作站)初始情况下没有(无法)配置 IP 地址,则需使用**逆向地址解析协议**(Reversed ARP,RARP),发送 RARP 广播(包含自己的 MAC 地址),从 RARP 服务器的响应中取得 IP 地址。RARP 如今已无应用场合。

ARP 易被网络攻击者利用,如进行 ARP 欺骗攻击。攻击计算机假冒某一台计算机的 IP 地址频繁进行 ARP 广播,使其他计算机误传信息给攻击者,或假冒路由器的 IP 地址进行 ARP 广播,将发往其他网络的报文引入歧途。直接的影响是计算机的正常通信被中断,或无法接入 Internet;间接的影响是由于报文发送受阻,计算机会反复发送 ARP 探询,导致网络中充斥广播报文,计算机也忙于更新地址映射表,占用了大量带宽和计算资源。

3.4.2　DHCP

动态主机配置协议(Dynamic Host Configuration Protocol,DHCP)为网络互连服务,却是一个应用层协议,1993 年 10 月定义为 RFC 2131/2132 标准。基于 IPv6 的标准 DHCPv6 在 RFC 3315 定义。DHCP 工作在 UDP 上(端口号服务端 67、客户端 68),主要用途是为接入网络的计算机自动分配 IP 地址、子网掩码、网关(路由器)IP 地址和 DNS 服务器地址(主、辅),避免用户手工配置的烦琐,也有利于 IP 地址的合理利用,防止 IP 地址冲突等问题,并可协助进行网络管理(如 ISP 对上网终端的管理)。

DHCP 采用 C/S 模式,是早期 BOOTP 的扩展。BOOTP 只能为计算机分配预置的静态 IP 地址,而 DHCP 的优势在于动态分配。DHCP 的报文结构如图 3.26 所示。

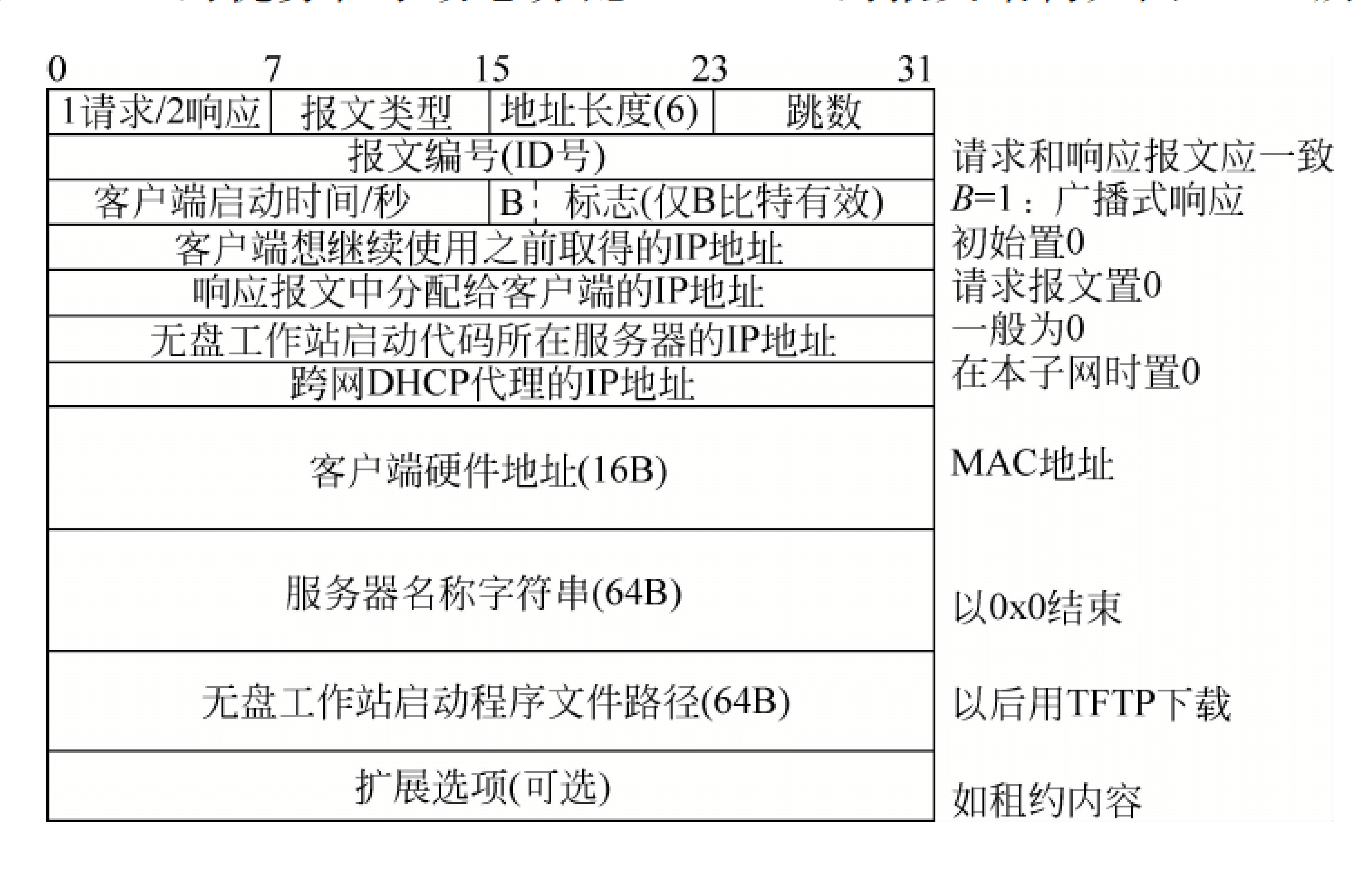


图 3.26　DHCP 报文格式

DHCP 报文分为客户端发送的请求报文和服务器发送的响应报文,共有 7 种报文类型,各自的功能结合如下的 DHCP 的四步工作流程具体阐述。

(1) 请求租约。当计算机首次连接网络的时候,广播 DHCP DISCOVER 报文,源 IP

地址设为 0.0.0.0,目的 IP 地址设为 255.255.255.255。报文的四次等待时间可预设为 1/9/13/16 秒。

(2) 提供租约。服务器从尚未租出的地址范围内,选择最前面的空闲 IP 地址,连同其他设定和租约期限,向客户端发送 DHCP OFFER 响应,报文编号与 DISCOVER 相同,源 IP 地址为服务器地址,目的 IP 地址为 255.255.255.255。IP 地址等设定值位于报文的扩展选项字段,包括子网掩码、网关地址、DNS 地址、租约期限等,分别采用＜type＞＜len＞＜value＞的编码结构。如果客户端要求的 IP 地址已经失效或已经被其他设备占用,服务器响应一个 DHCP NACK,要求其重新执行 DHCP DISCOVER。

(3) 选择租约。如果客户端收到多台 DHCP 服务器的响应,则须选择其中一个 DHCP OFFER(通常是最先抵达的那个),并广播 DHCP REQUEST 报文,以告知所有 DHCP 服务器。同时,客户端采用 ARP 查询该 IP 地址是否已被占用,如果已被占用,则发送 DHCP DECLIENT 给服务器,拒绝接受 DHCP OFFER,并重新发送 DHCP DISCOVER。

(4) 确认租约。当 DHCP 服务器接收到 DHCP REQUEST 后,向客户端发出 DHCP ACK 响应,确认 IP 地址租约正式生效。若客户端想退租,可以随时发送 DHCP RELEASE 命令解约。

如果 DHCP 租约超过一定时间(如过半),客户端可发送 DHCP REQUEST,以保持与服务器的联系;假如租约快到期时得不到服务器响应,客户端将尝试联络其他 DHCP 服务器。

如果 DHCP 服务器位于其他网络(网段),由于客户端尚未设定 IP 参数,不知道路由器地址,大多数路由器也不会转发 DHCP 广播报文,因此,需要采用 DHCP 代理(DHCP Agent/Proxy),将请求传递给 DHCP 服务器,并将服务器响应转发给客户端。

3.4.3 ICMP

Internet 控制消息协议(Internet Control Message Protocol,ICMP)是 TCP/IP 的核心协议之一,用于发送控制消息,获取可能发生在通信环境中的各种问题的反馈信息,令管理者可以对问题做出诊断。

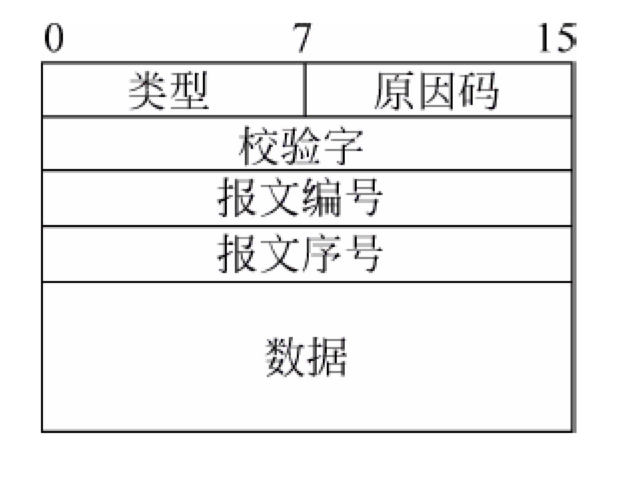


图 3.27 ICMP 报文格式

ICMP 是网络层的子层协议,使用 IP 封装(协议字段值为 1),在 RFC 792 标准中定义。ICMP 报文结构如图 3.27 所示。

ICMP 报文类型较多,常用的有:类型 8/0 回显请求和响应;类型 3 目标不可达(原因码有 0/1/2/3 目标网络/主机/协议/端口不可达等);类型 11 超时(原因码 0 为 TTL 超时,原因码 1 为分片组装超时);类型 13/14 时间戳请求/响应;类型 17/18 网络掩码请求/响应;类型 30 路由跟踪请求;等等。

网络诊断中有两个常用的软件工具:traceroute 发送包含有特殊 TTL 的 ICMP 路由跟踪报文,根据接收到的超时消息和目标不可达消息来实现,可查看从源到目的的各级路由器信息;ping 则通过 ICMP 回显报文(请求和响应)来实现,可选择发送各种长度的数据(位于报文数据字段),目的方接收到报文后,将数据原封不动地在响应报文中返回,称为一次回显(echo)。根据多次回显信息可以判断:目标的域名和地址是否正确、目标是否可达、目的计算机是否在运行、报文来回时间多少以及是否稳定、信道质量如何(是否丢失报文)等。

ECHO 是希腊神话中森林女神的名字，因触犯了赫拉，被诅咒只能重复她听到的话。从此，人们在山谷里呼唤，就会听到远远传来的 ECHO 的重复声。

3.4.4　IGMP

当一台主机需要发送相同信息（如视频会议画面）给一组（设为 N 台）目的计算机，若采用单播的方式，则每个报文都需要发送 N 次，发送方和网络都要承受较大压力，若采用广播方式，则会影响其他非接收方计算机，而且跨子网（或自治域）时并不支持广播。

比较图 3.28 可以看出，若采用**组播**（Multicast）方法（有些文献称为多播），可以将网络流量压缩到最低程度，显著减少网络资源消耗，提高网络利用率，而且，一组组播流便于采用一个服务质量（QoS）策略进行控制。因此 IGMP 属于一种重要的子网带宽管理（Subnet Bandwidth Management，SBM）方法。

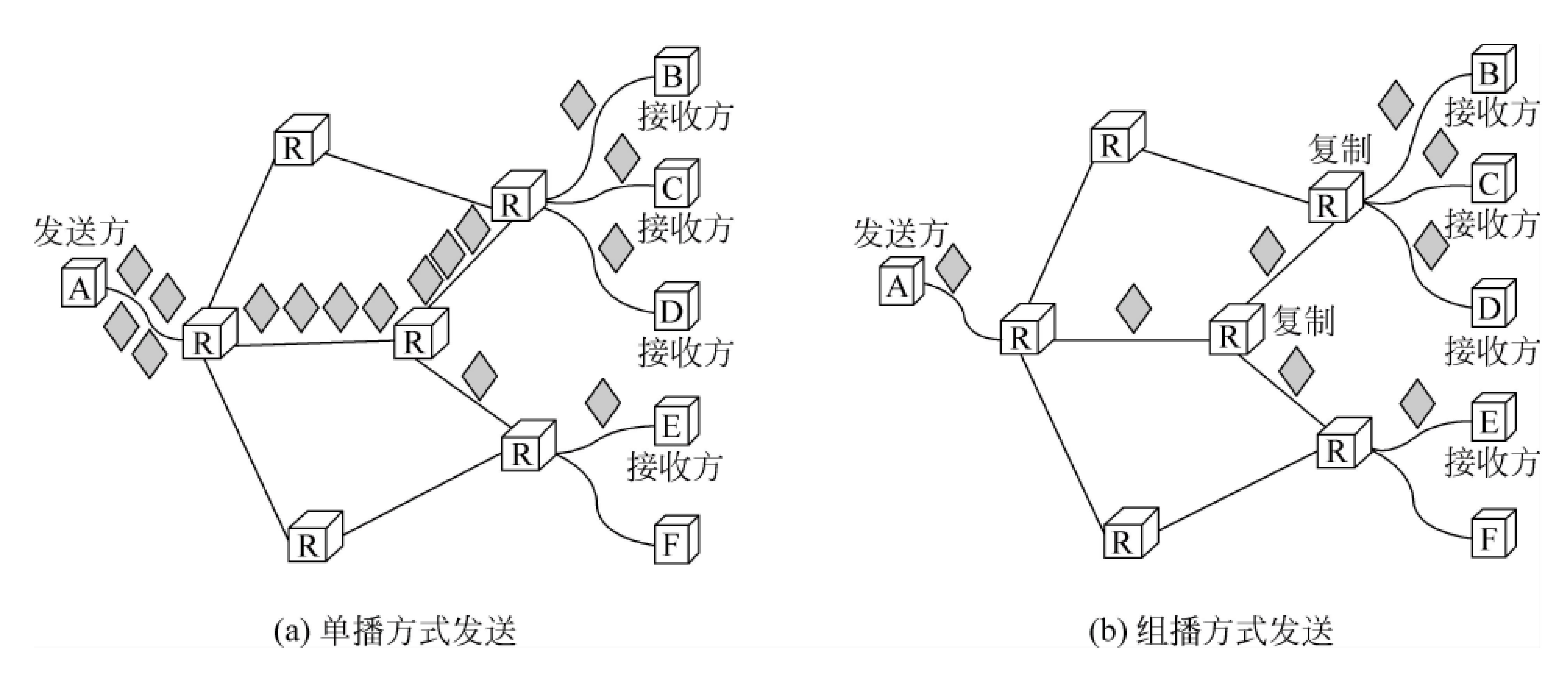


图 3.28　组播和单播方式流量比较

组播采用 IP 地址空间中的 D 类地址，前缀为 1110，地址范围为 224.0.0.0～239.255.255.255。每个 D 类地址都可标识一组主机（而非单播地址仅标识一台主机），即最多有 2^{28} 个主机组（Host Group）或称组播组。如果向一个 D 类地址发送一个数据报文，就相当于向这个组播组中的所有主机发送同样的报文。可见，组播地址只能用于目的地址，不能用于源地址，并且 IP 报文仍然采用尽力而为的交付方式，不保证交付的可靠性。

其中有一些 D 类组播地址已经被 IANA 指定为永久组地址，具有网络操作上的特殊含义或限定。

（1）224.0.0.0：基地址（保留）。

（2）224.0.0.1：在本子网上的所有参加组播的主机和路由器。

（3）224.0.0.2：在本子网上的所有参加组播的路由器。

（4）224.0.0.3，224.0.0.19～224.0.0.225：未指定。

（5）224.192.0.0～239.251.255.255：限制在一个组织的范围内。

（6）239.252.0.0～239.255.255.255：限制在一个地点的范围内。

特别地，在 Ethernet 上，可利用 Ethernet 的 MAC 地址进行组播操作，因此，就需要将 IP 组播地址与 MAC 组播地址进行对应转换（如图 3.29 所示）。由于 IANA 拥有的 MAC 组播地址范围为 00-00-5E 为前缀的一半地址空间，加上 MAC 组播地址的规定（第一字节

最低位为 1),即组播地址范围从 01-00-5E-00-00-00 到 01-00-5E-7F-FF-FF,所以 MAC 组播地址中仅有 23b 可与 IP 组播地址的 28b 对应,有 5b 无法复制,这样,不可避免地会出现多个 IP 组播地址对应到同一个 MAC 组播地址上,需要接收方主机对报文进行软件过滤。

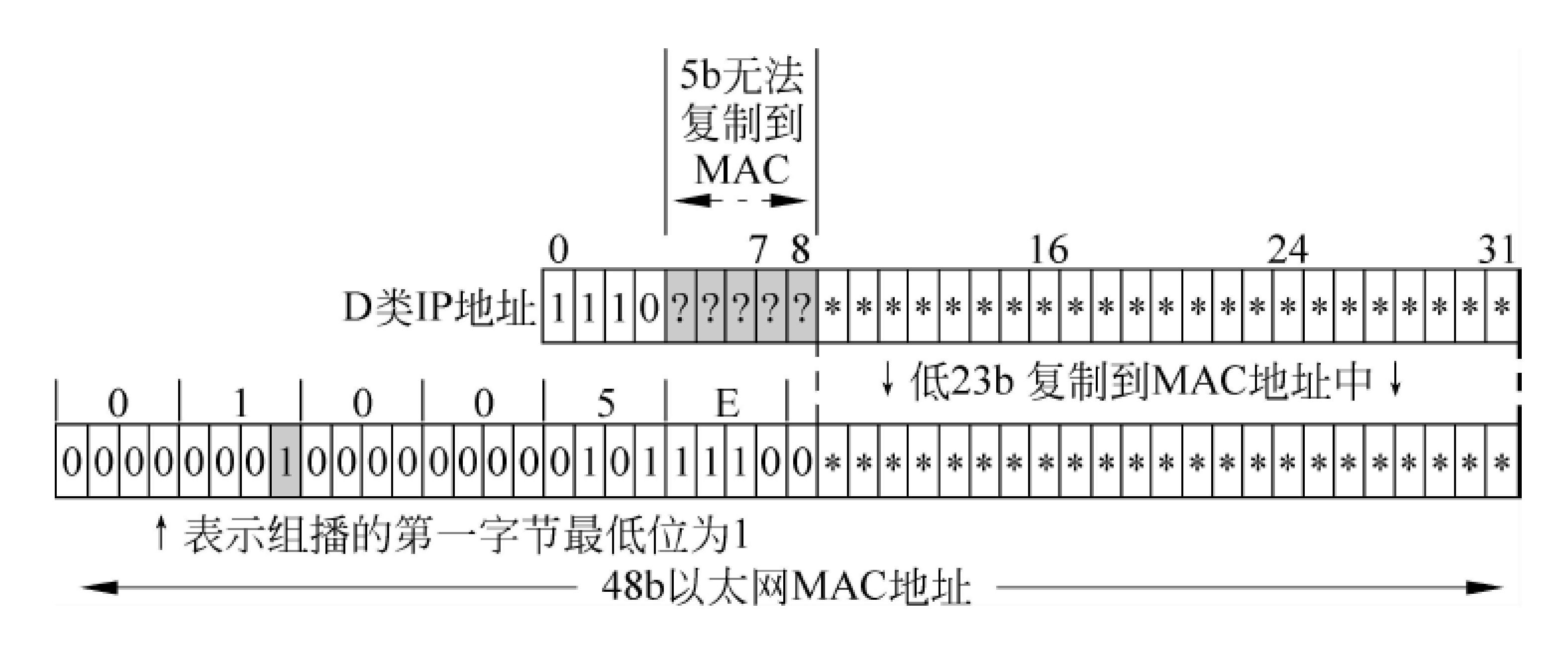


图 3.29 IP 和 MAC 的组播地址映射关系

组播组中的成员(主机)是动态的。一个主机可以按照一定方式加入某个组播组。加入后,主机就向组播地址(即面向所有成员)发送报文,声明其组员关系。组播路由器收到此声明后,进一步通知其他组播路由器。组播组的大小没有明确限制,可以达到成千上万个,但是很明显,维护组播组需要较大的开销,而且所有接收主机所在的转发路径上的路由器都应支持组播功能,才能真正达到组播的目的。

互联网组管理协议(Internet Group Management Protocol,IGMP)就是实现 Internet 组播管理的技术,现使用版本 2(RFC 2236)。IGMP 协助路由器识别加入到一个组播组的成员主机。

IGMP 使用 IP 传递报文(协议字段值为 2),同时反过来为 IP 提供服务,因此 IGMP 应作为网络层的一个子层来看待。



图 3.30 IGMP 报文格式

IGMP 的报文格式如图 3.30 所示。

长度为 8B 的 IGMP 报文分为 4 个字段。

(1) 类型字段,指出 IGMP 报文的类型(如表 3.8 所示),有一种 IGMPv1 报文和三种 IGMPv2 报文。

(2) 响应时间字段,以 1/10s(100ms)为单位,默认值为 10s。

(3) 校验字字段,对整个 IGMP 报文进行校验,校验算法与 IP 报文相同。

表 3.8 IGMP 报文类型

类型	组地址	报 文 含 义	版本
0x11	全 0	一般的成员关系询问	v2
	指定	特定的成员关系询问	v2
0x16	指定	成员关系报告	v2
0x17	指定	离开组	v2
0x12	指定	成员关系报告	v1

(4) 组地址字段，在对所有的组发出询问时填充全零，对特定的组询问时填入该组的组地址，主机发送成员关系报告时填写自己的组地址。

显然，IGMP 的执行离不开 IP 的配合。不仅因为 IGMP 需要 IP 的传输，更重要的是，当一个主机用 IGMP 报告成员关系（或离开组）时，其他主机和路由器将依据携带该 IGMP 报文的 IP 报文中的 IP 源地址来确定谁在报告加入组（或离开组）。

路由器通过 IGMP 获知每个组播组中的成员主机。如果一些主机在路由器直接互连的子网内，路由器将把接收到的组播报文复制后依次传送给这些主机（或通过以太网组播进行发送）；如果还有参与组播的主机不在直接互连的子网内，则根据路由信息，把组播报文复制后传送给相关的路由器（可能分布在多个路由器上）。

然而，确定和维护组播路由是相当复杂的，包括可能需要穿越没有任何组成员的网络或不支持组播的网络（需采用隧道技术）。一旦一个组成员加入或离开，都可能导致组播路由发生较大变化，并且需要相当的资源开销来通知和实现这一变化。

Internet 上采用的典型组播路由选择协议是距离向量组播路由选择协议（Distance Vector Multicast Routing Protocol，DVMRP），由 RFC 1075 定义。同类协议还有：核心基干树（Core Based Tree，CBT），RFC 2189/2201；OSPF 组播扩展（Multicast Extensions to OSPF，MOSPF），RFC 1548；协议无关组播-稀疏/密集方式（Protocol Independent Multicast-Sparse/Dense Mode，PIM-SM/DM），RFC 2362 等。

思考：分析各个成员在网络上不同的分布情况下对组播技术执行效能的影响。试用组播协议设计或改进一个群聊系统。

3.4.5 SNMP

网络通信系统是个复杂的综合体，包括智能、非智能的网络设备、通信媒介、各类计算机、外设、固件（firmware）、系统软件、应用程序等。尤其是一些大型的信息系统，覆盖范围广，维护和管理依赖人力必然难以胜任。因此，**网络管理**（Network Management）和**系统管理**（System Management）成为网络和信息系统不可或缺的组成部分，也是安全保障的工具之一。

网络管理和系统管理在管理对象上略有差异，系统管理的范围稍宽，除了网络管理应承担的职能外，还包含外设资源、文件资源、应用（软件）资源、数据资源等方面，但两者在主要目的和技术原理、方法上是一致的。

网络管理可以保障系统稳定、安全、高效地运行；可提供集中式的、全面的监测（survey）和控制（control）手段，有利于对整个系统进行有效的管理和维护；可及时、准确地发现系统故障、定位故障点，并提醒管理人员尽快修复；可通过对系统运行数据的采集和分析，为系统优化和改进提供依据。

网络管理通过网络管理系统来实现。如图 3.31 所示，网络管理系统一般由相互关联的 4 个方面组成。

(1) 管理者（Network Manager），或称管理中心，是网络管理的主体。

(2) 被管理者或称管理对象（Managed Object），是网络系统中所有软件和硬件设备的集合，通过管理代理（Managed Agent）与管理者通信。

(3) 管理方法，是实施网络管理的技术方案、系统结构，核心是管理协议和**管理信息库**

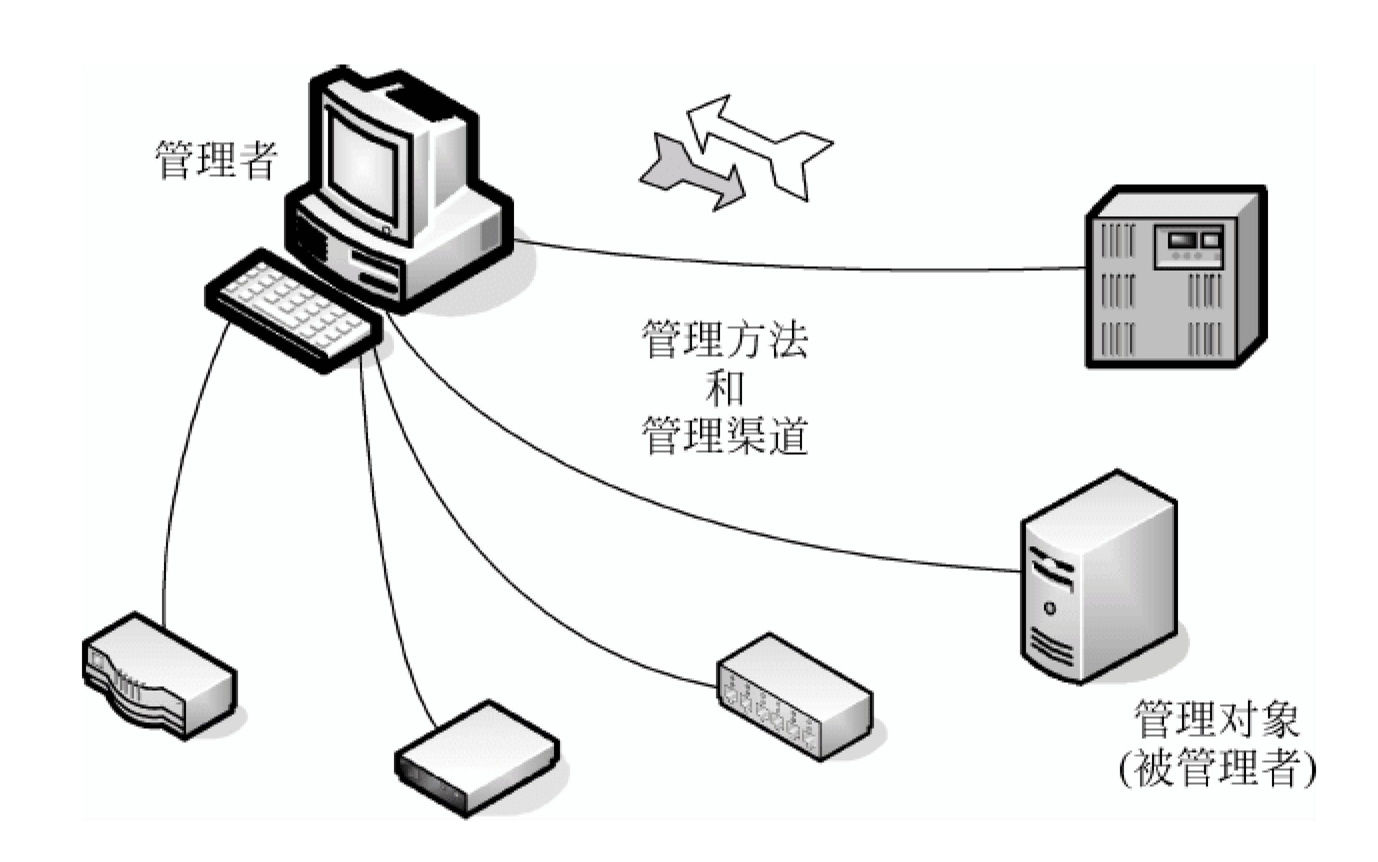


图 3.31　网络管理体系结构概念

(Management Information Base,MIB)。

(4) 管理渠道,是管理者和管理对象间交换数据的载体。

网络管理系统与其管理的网络系统有着密不可分的联系。网络管理系统一般运行在网络系统之上(带内通信方式),一定程度上依赖网络系统提供互连服务;网络管理系统也需要网络系统提供管理所需的运行数据;同时,网络管理系统又是网络系统的监管者,是管理者和被管理者的关系。因此,处理好网络管理系统和网络系统之间的关系显得至关重要。

ISO 建议网络管理应包含以下 5 个基本功能域:故障管理、配置管理、性能管理、安全管理和计费管理。

(1) **故障管理**(Fault Management)。故障管理是网络管理中最基本的功能之一。当网络发生故障时:①必须尽可能迅速地找出故障发生的确切位置;②将网络其他部分与故障部分隔离,以确保网络其他部分能不受干扰继续运行;③重新配置或重组网络,尽可能降低由于隔离故障后对网络带来的影响;④修复或替换故障部分,将网络恢复为初始状态。对网络组成的部件状态的监测是网络故障检测的依据。不严重的简单故障或偶然出现的错误通常被记录在错误日志中,事后做出处理;而严重一些的故障则需要通知网络管理器,即发出报警。因此网络管理器必须具备快速和可靠故障监测、诊断和恢复功能。

(2) **配置管理**(Configuration Management)。配置管理也是网络管理的基本功能。计算机网络由各种物理结构和逻辑结构组成,这些结构中有许多参数、状态等信息需要设置并协调。另外,网络运行在多变的环境中,系统本身也经常要随着用户的增、减或设备的维修而调整配置。网络管理系统必须具有足够的手段支持这些调整的变化,使网络更有效地工作。这些手段构成了网络管理的配置管理功能。配置管理功能至少应包括识别被管理网络的拓扑结构、标识网络中的各种事件、自动修改指定设备的配置、动态维护网络配置数据库等内容。

(3) **性能管理**(Performance Management)。性能管理的目的是在使用最少的网络资源和具有最小延迟的前提下,确保网络能提供可靠、连接的通信能力,并使网络资源的使用达到最优化的程度。网络的性能管理有监测和控制两大功能,监测能实现对网络中的活动进行跟踪,控制功能实施相应的调整来提高网络性能。性能管理的具体内容包括:从被管对象中搜集与网络性能有关的数据,分析和统计历史数据,建立性能分析的模型,预测网络性

能的长期趋势，并根据分析和预测的结果，对网络拓扑结构、某些对象的配置和参数做出调整，逐步达到最佳运行状态。如果需要做出调整时，还要考虑扩充或重建网络。

(4) **安全管理**(Security Management)。安全管理的目的是确保网络资源不被非法使用，防止网络资源由于入侵者攻击而遭受破坏。其主要内容包括：与安全措施有关的信息分发(如密钥的分发和访问权限设置等)，网络安全通知(如网络有非法侵入、无权用户对特定信息的访问企图等)，安全服务措施的创建、控制和删除，与安全有关的网络操作事件的记录、维护和查询日志管理工作等。一个完善的计算机网络管理系统必须制定网络管理的安全策略，并根据这一策略设计实现网络安全管理系统。

(5) **计费管理**(Accounting Management)。在商业有偿使用的网络上，计费管理功能表现为：一方面，统计哪些用户、使用何信道、传输多少数据、访问什么资源等信息；另一方面，计费管理功能还可以统计不同线路和各类资源的利用情况。因此可见，计费管理的根本依据是网络用户的使用统计信息，并制定一种用户可接受的计费方法。商业性网络中的计费系统还要包含诸如每次通信的开始和结束时间、通信中使用的服务等级，以及通信中的另一方等更详细的计费信息，并能够随时查询这些信息。

一个具体的网络管理系统不一定要完全包含上述的五大管理功能域，不同的系统可以选取其中的几个功能域或功能域中的部分功能加以组合。但几乎每个网络管理系统都会包括故障管理功能。

如图 3.32 所示，网络管理系统利用管理代理不断采集网络运行的各种信息，存储在管理代理维护的本地 MIB 中。当网络管理者需要时，管理代理将交付最新的 MIB 数据。管理代理时刻监听来自网络管理者的查询或命令，并及时做出响应。命令都由网络管理者发出，get 命令用于监视，即获取数据；set 命令用于控制，即设置参数。还有一类报文 trap，是由管理代理主动发起，用于故障情况的告警(alert)。

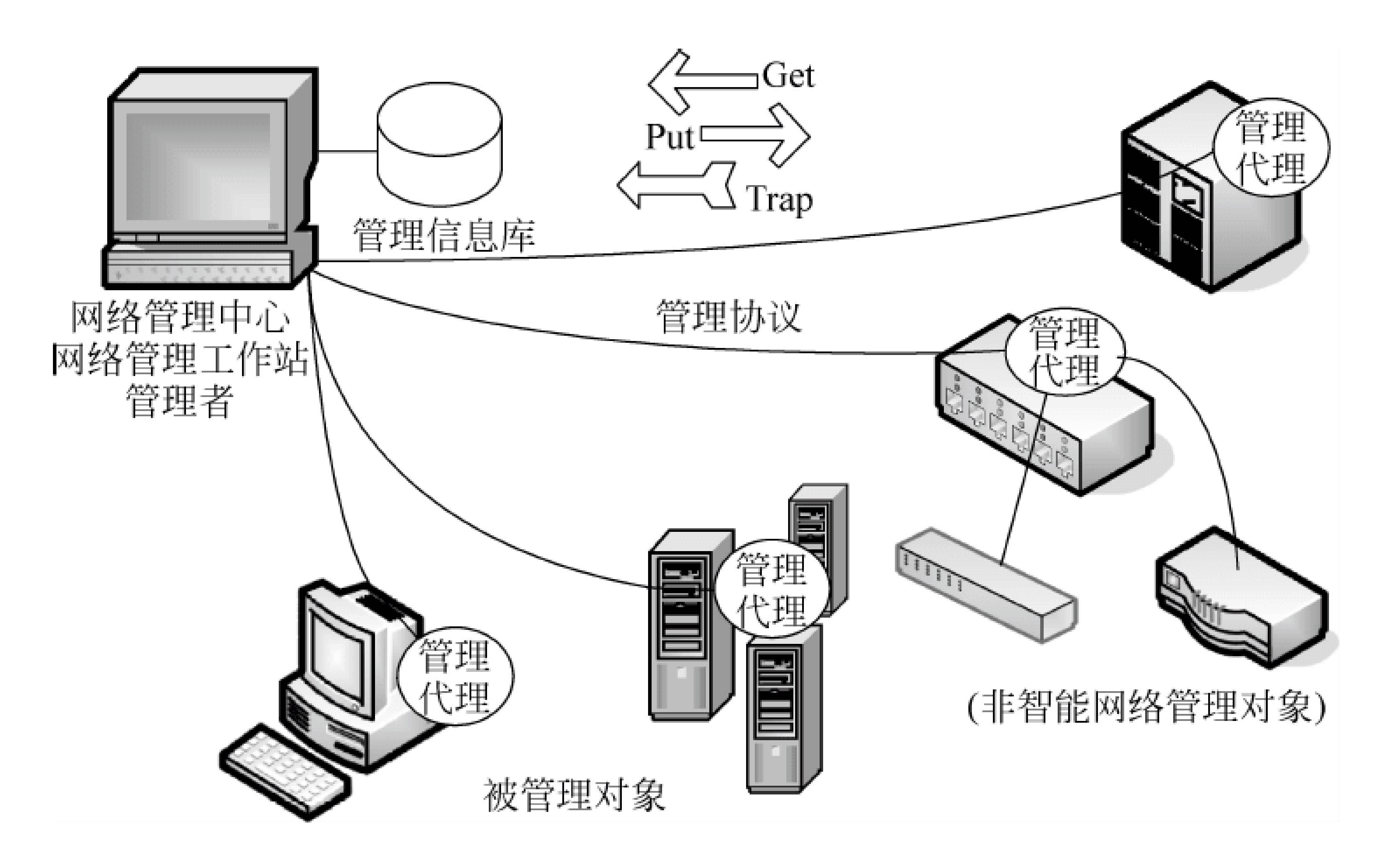


图 3.32　网络管理系统结构示意

最常用的网络管理协议有两个：Internet 于 1988 年提出的**简单网络管理协议**(Simple Network Management Protocol，SNMP)、ISO 提出的**公共管理信息服务**(Common Management Information Service，CMIS)和**公共管理信息协议**(Common Management Information Protocol，CMIP)。

与 CMIS/CMIP(ISO 7498-4/ITU-T X.700)相比,SNMP(RFC 1155,RFC 1156//1157)以其简易性与可扩展性的优势得到更广泛的应用。SNMP 体现了网络管理系统的一个重要准则,即网络管理功能的实现不能影响网络系统的正常功能,不给网络附加过多的开销。SNMP 还由 RFC 1441～1452 定义了第二版,由 RFC 3411～3418 定义了第三版。

SNMP 是一种基于 UDP 的应用层协议,管理工作站根据管理需要产生三种类型的 SNMP 消息:get request(类型 0)、get next request(类型 1)和 set request(类型 3),代理均以 get response(类型 2)确认,或主动发送 trap(类型 4)。

SNMP 报文结构如图 3.33 所示。版本号字段为 SNMP 版本号减 1(目前最高为 v3,则填 2);共同体字段为管理者与代理之间的明文口令,默认为 6 个字符的 public;请求标识符字段用于区分同时进行的多个 get 任务,在 get request 和 get response 中应一致;差错状态字段表明变量不存在等情况,并通过差错索引字段指出变量的偏移值,0 表示正常;trap 类型字段描述代理初始化、接口故障与恢复等状况;企业标识字段为该设备的生产商在对象命名树的 enterprises 结点的子树上的编码;特定代码字段为自定义事件,一般为 0;时间戳字段指明代理初始化到 trap 报告事件发生的间隔(以 10ms 为单位)。

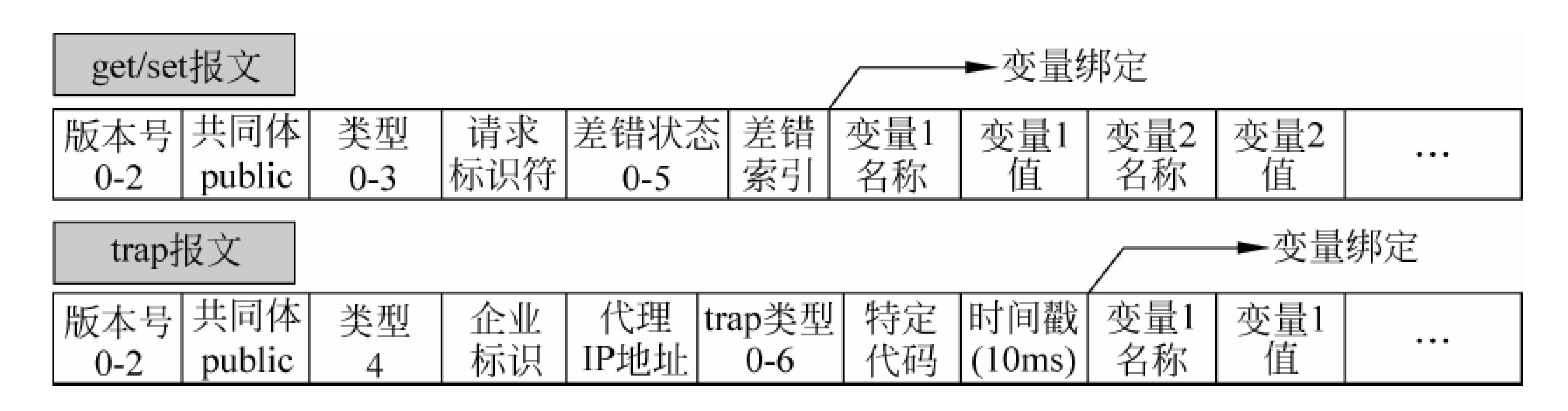


图 3.33 SNMP 报文格式

SNMP 的 MIB 是网络中所有被管对象的数据集合,对象应是对象命名树(Object Naming Tree)上的一个结点(如图 3.34 所示)。例如,Internet 对象为 1.3.6.1,mib2 中的

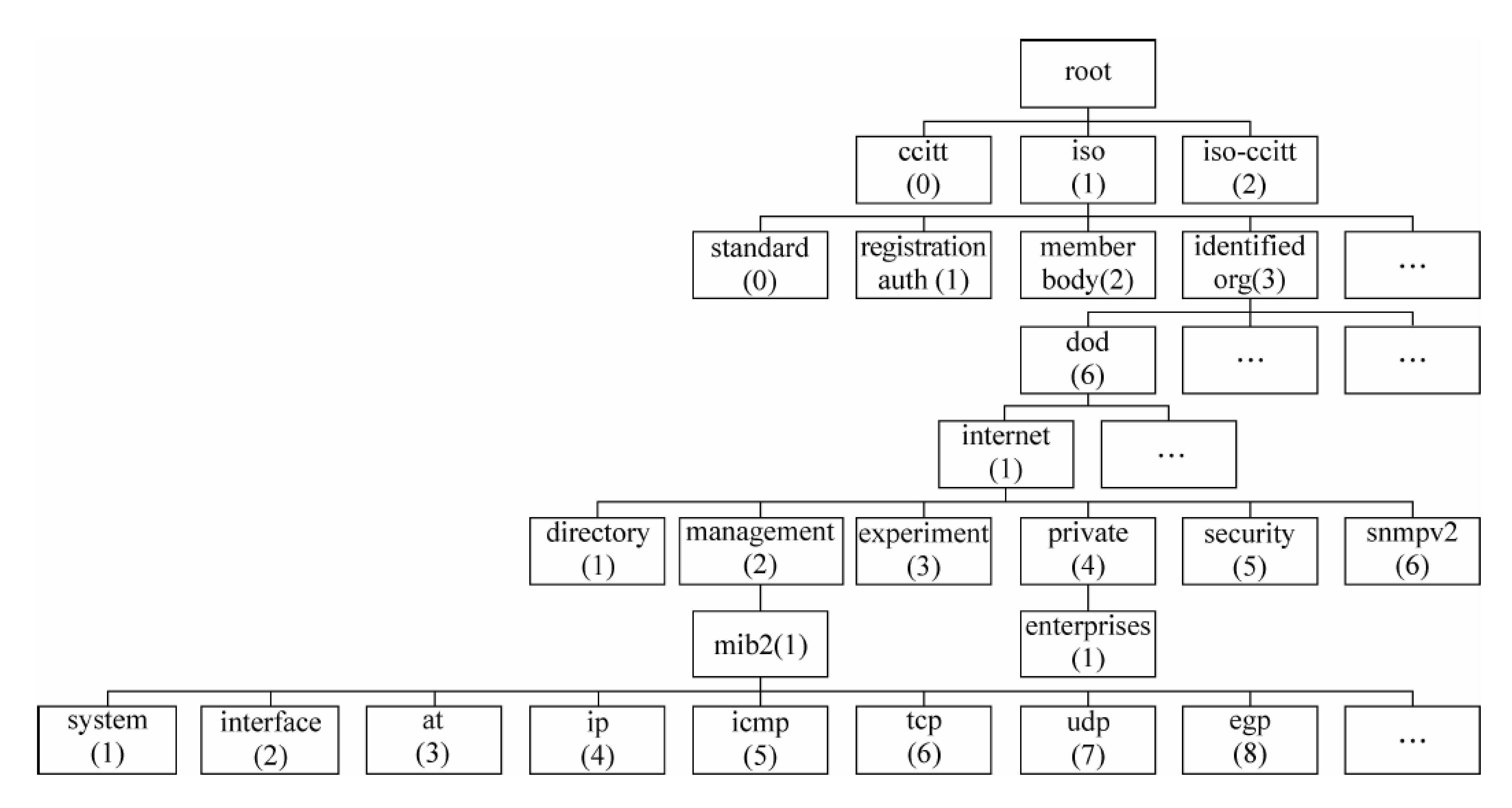


图 3.34 MIB 对象命名树

IP 对象为 1.3.6.1.2.1.4。而变量则是特定对象的一个实例。例如，ifNumber 为网络接口数，ipInReceives 为接收到的 IP 报文数，tcpMaxConn 为允许的最大 TCP 连接数等。变量及其数值采用抽象语法记法 ASN.1 定义。

SNMP 采用一种不完全的轮询协议，一方面通过依次探询各个代理，获取网络管理信息，另一方面也允许未经询问发送报告(trap)。

网络管理技术不断在完善发展中，技术更先进、部署更灵活、功能更丰富、管理更细致、适用更广泛、操作更便捷，技术标准之间也在逐步融合，如 CMOT(CMIP Over TCP/IP)就同时适用于 TCP/IP 和 OSI 网络环境。

CHAPTER 4

第 4 章 Internet 路由协议

4.1 Internet 路由原理

IP 网络的**路由**(routing)由路由器(router)完成。路由器也称网关(gateway),主要实现两个方面的功能:

(1) 为各个网络间的数据交换寻找最佳路径;

(2) 为给定的 IP 报文提供到指定目的地的转发服务。

如图 4.1 所示,设一台计算机(源地址为 IP_{src})发送 IP 报文到目的地址 IP_{dst}。如果 IP_{src} 和 IP_{dst} 在同一网段中,可在 IP-MAC 地址映射表中检索到,则通过二层网络(如 Ethernet)直接送达;如果不属于同一网段,则必须交由路由器转发。路由器根据路由信息,把 IP 报文转发给下一个网关,直到到达目的计算机所在的子网。

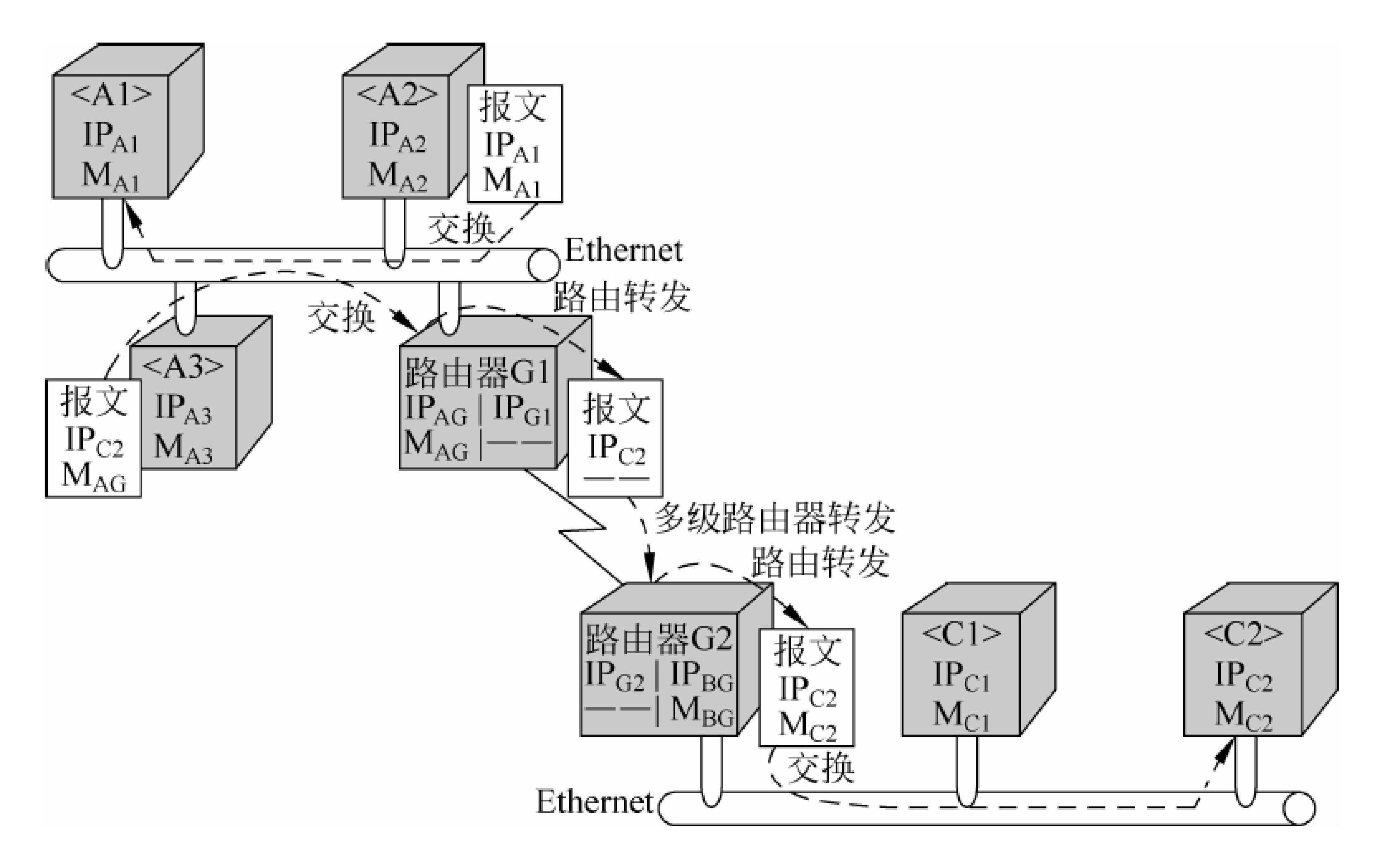


图 4.1 IP 报文路由与转发示意

路由是一种分布式控制技术，可以自动实现，而且可以根据网络拓扑结构的变化动态调整。路由的建立需要采用特定的路由算法，通过路由协议的执行，路由器完成相互间的路由信息交换，最终生成路由表。每台路由器均维护一张路由表，由{目的网络(主机)地址；跳数；下一跳地址}格式的记录构成。报文转发就是根据目的地址检索路由表，然后发送给下一跳(next hop)地址指定的路由器。因此，**一个 IP 报文并没有被事先安排好发送路径，而是由途经的每一台路由器独立决定如何发送**。虽然如此，IP 报文的路径仍然是可预测的。

路由算法应按照特定的策略(policy)来设计。如图 4.2～图 4.4 所示，在相同的网络环境下，如果分别选择最少跳数、最短距离、最大带宽的策略，得到的路径各不相同。此外，还可以采用一些辅助策略，例如在线路拥塞、故障状态下，改由后备路径转发，以避免拥塞的加剧或无法及时转发报文。

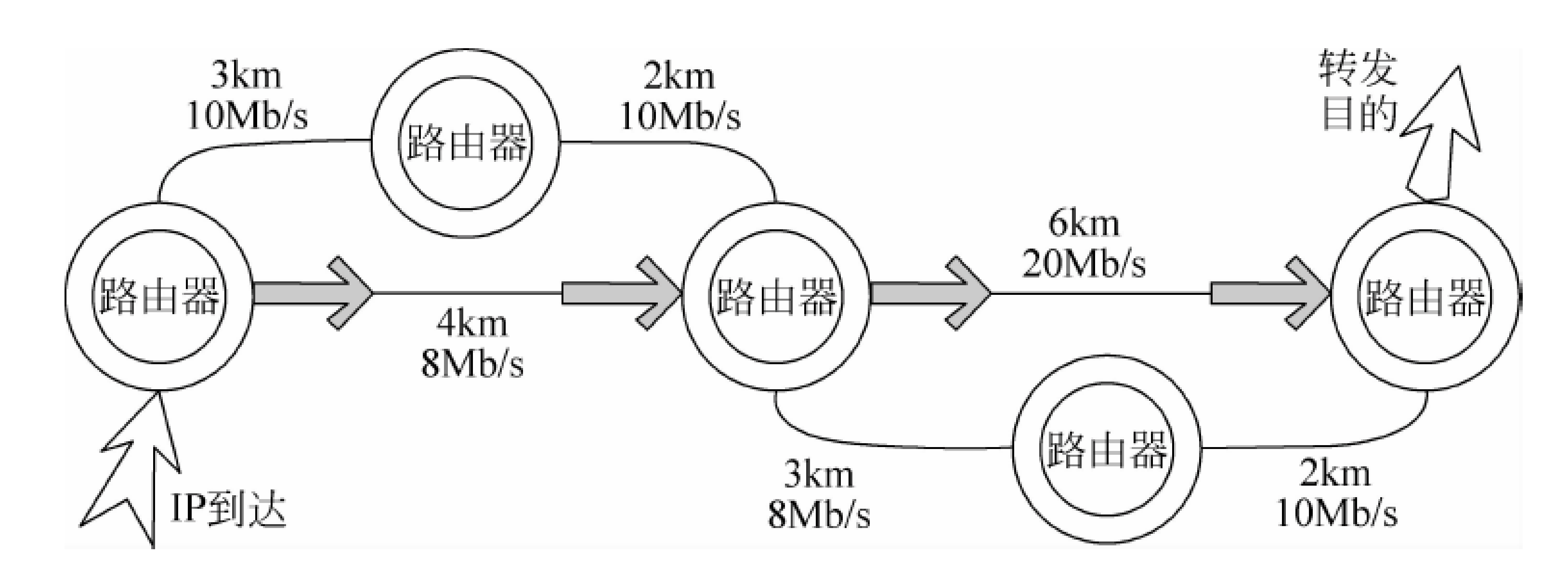


图 4.2　最少跳数路由选择策略

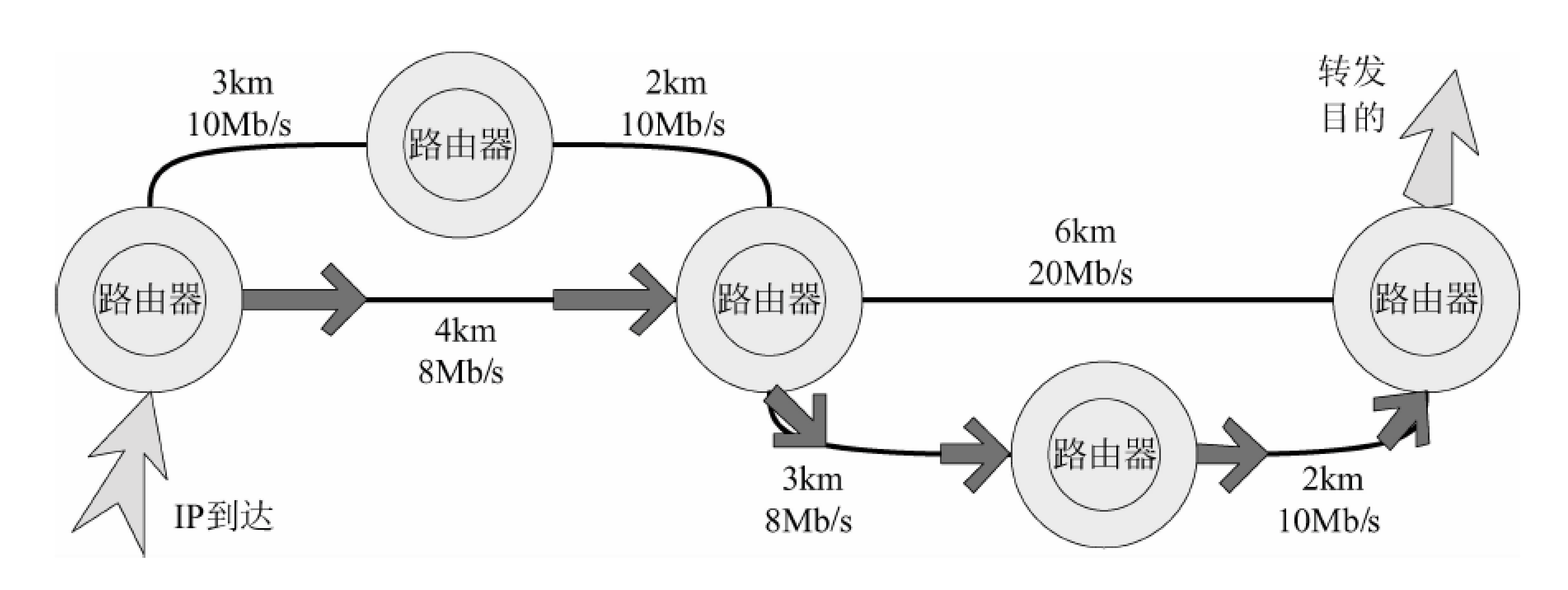


图 4.3　最短距离路由选择策略

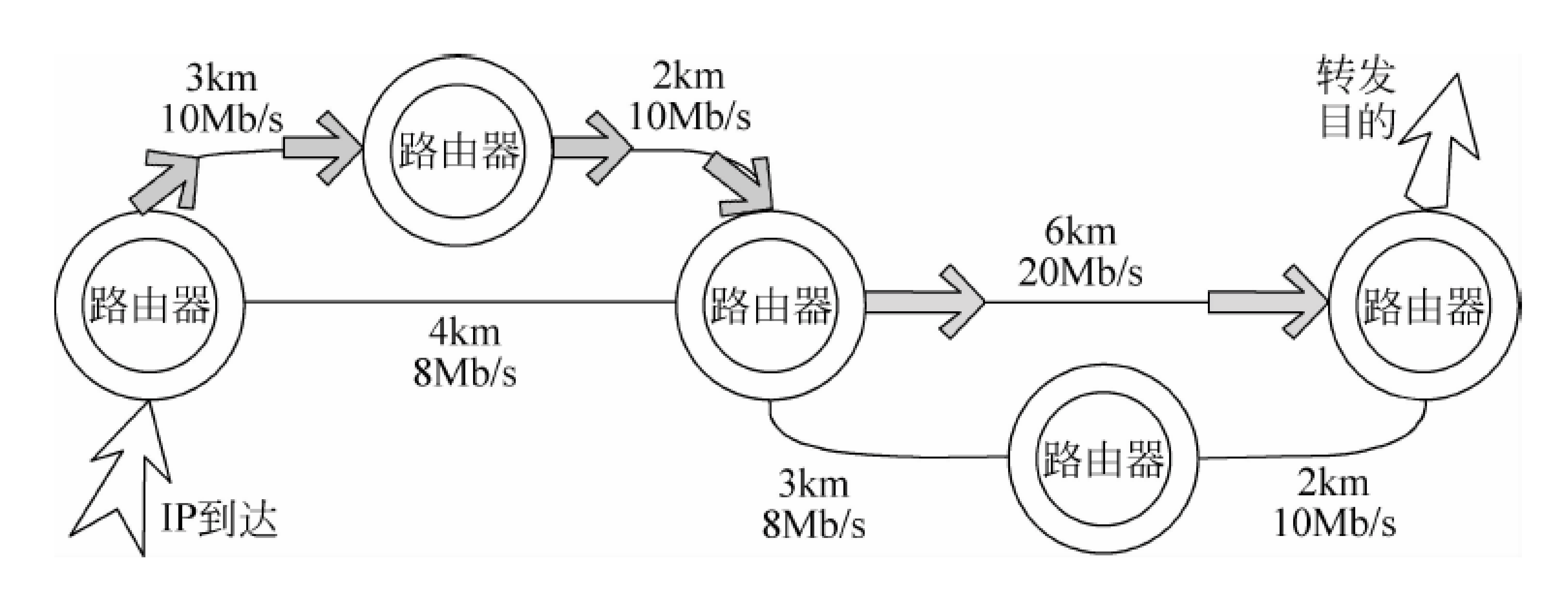


图 4.4　最大带宽路由选择策略

路由策略可由**自治系统**(Autonomous System,AS)自行决定。自治系统是指由一个实体管理的IP网络,有相对独立性,其内部采用同一种路由算法、同一个AS编号,并与其他AS互连。有些路由算法和协议适合用户网络接入AS,有些路由算法和协议用于AS内部,还有些路由算法和协议适用于AS之间的互连,最终构成完整的Internet路由体系。

思考:*路由和交换有何共同点和不同点?*

4.2 Internet路由协议概述

路由算法是路由协议的灵魂。一个好的路由算法可以快速建立和更新路由信息,并对各种网络故障或改变做出快速响应。因此,路由算法的设计需要综合考虑以下性能目标。

(1) 完整性:到达任一结点的报文,都能根据路由信息进行转发,最终到达报文指定的目的地。

(2) 最优性:路由算法具有根据既定策略选择最佳路径的能力。

(3) 简洁性:算法设计简洁,用最少的软件开销,提供最有效的功能,并且尽可能降低时延。

(4) 坚固性:路由算法处于非正常或不可预料的环境时,如硬件故障、负载过高或操作失误,仍能保持正确运行。由于网关分布在网络连接点上,因此路由故障的后果很严重。路由算法应能经受时间的考验,在各种网络环境下被证实是可靠的。

(5) 收敛性:收敛是在最佳路径的判断上所有网关达到一致的过程。当某个网络事件引起路由可用或不可用时,相关网关就发出更新信息。路由更新信息传遍整个自治域网络,引发重新计算最佳路径,最终达到一致公认的最佳路径。收敛慢的路由算法可能造成暂时的路径循环,或导致网络中断。

(6) 灵活性:路由算法可以快速、准确地适应各种网络环境。例如,一条线路发生故障,路由算法应能很快发现故障,并选择另一条最佳路径。但算法结果应保持稳定,不能频繁变化。

路由算法可依据的度量(metric)有链路长度、带宽、吞吐量、保密要求、传播时延、某时段内的通信流量、缓存被占用的程度、链路差错率等。

路由算法有静态路由和动态路由之分。前者为固定配置路由,无更新信息的开销,但无法适应变化;后者的更新虽然需要占用网络资源、有一定滞后,但可以自动调整路由信息以适应网络状况。

依据路由算法和协议应用的网络位置,可分为两种类型。

(1) **内部网关协议**(Interior Gateway Protocol,IGP),在AS内部范围使用,如RIP、OSPF,实现域内路由选择(intradomain routing)。

(2) **外部网关协议**(External Gateway Protocol,EGP),在AS之间使用,如BGP,实现域间路由选择(interdomain routing)。

Internet路由算法采用分布式的分层路由的思想,一个结点仅需关心和了解两方面的信息:与其直接连接的结点、AS内的其他结点。除此以外的结点信息,这个结点既无须了解,也无力全部掌握。AS以外的路由信息,由边界上的结点搜集,为所在的AS提供对外的路由。

4.2.1　RIP

路由信息协议(Routing Information Protocol,RIP)是 Internet 的内部网关路由协议(RFC 1058)。RIP 基于距离矢量算法,通过计算抵达目的地的最短距离来选取最佳路径。

每台路由器需要维护到 AS 内每一个目的网络的距离记录,称为距离矢量。如果路由器与网络直接连接,则定义距离为 0,可以进行报文直接交付;如果路由器经过 n 台路由器与目的网络间接连接,则定义距离为 n,每经过一台路由器被称为一跳(hop),因此对于 RIP 而言,距离就是跳数。

RIP 的跳数最多为 15 跳,当超过 15 跳时,RIP 会认为目的地不可达,所以 RIP 适用于小规模的网络(小型 AS)。

RIP 试图在路由器和目的网络之间找到一条最短距离(最少跳数)的路径,即最佳路径。最佳路径是唯一的。

RIP 每隔一个固定周期(如 30s)与且仅与相邻的路由器交换路由信息。交换的路由信息内容为已知的全部路由信息(即自己的路由表)。每台路由器的路由表就在相互交换信息的过程中不断完善,直至**收敛**(convergence),最终所有路由器都获得正确、完整的路由表。

RIP 以寻找最短路径即最少跳数的路由为目标,**Bellman-Ford 算法**(或 **Ford-Fulkerson 算法**)是其算法基础:

若 X 是 $A \rightarrow B$ 最短路径上的结点,则 $A \rightarrow X$ 和 $X \rightarrow B$ 都是最短路径。

设路由表的记录项表示为[目的地址,最短距离,下一结点]。

如图 4.5 所示,“目的地址”为报文所指定的目的网络的地址,实际上就是网络号(可以通过报文的目的地址和子网掩码相与获得);“最短距离”为从本结点开始到达目的网络所需经过的最少跳数;“下一结点”为产生最短距离的路径所经过的相邻结点的距离。

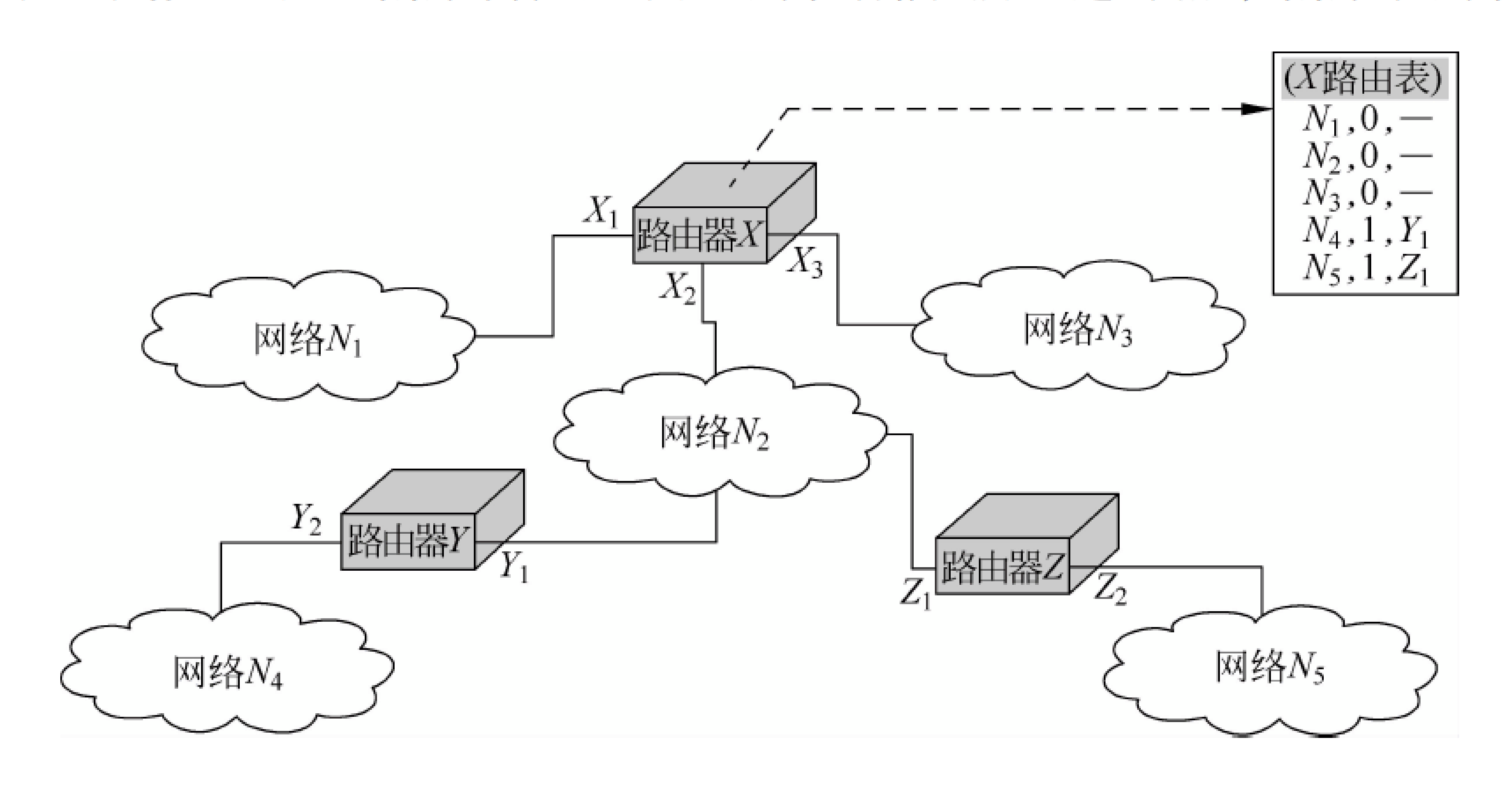


图 4.5　路由表路由信息构成示意

RIP 距离矢量算法如下。

初始情况下,若结点(路由器)直接连接网络 N_i,则生成相应的路由表记录项[N_i,0,—],其中“—”表示不需要经过其他结点。如果直接连接多个网络,则同理生成多条路由表记录。然后结点将自己的初始路由表发送给所有相邻结点。

当一个结点收到来自相邻结点(设其地址为 X_j)的 RIP 报文(其中包含了 X_j 的路由表)后,执行如下过程(参考如图 4.6 所示的实例)。

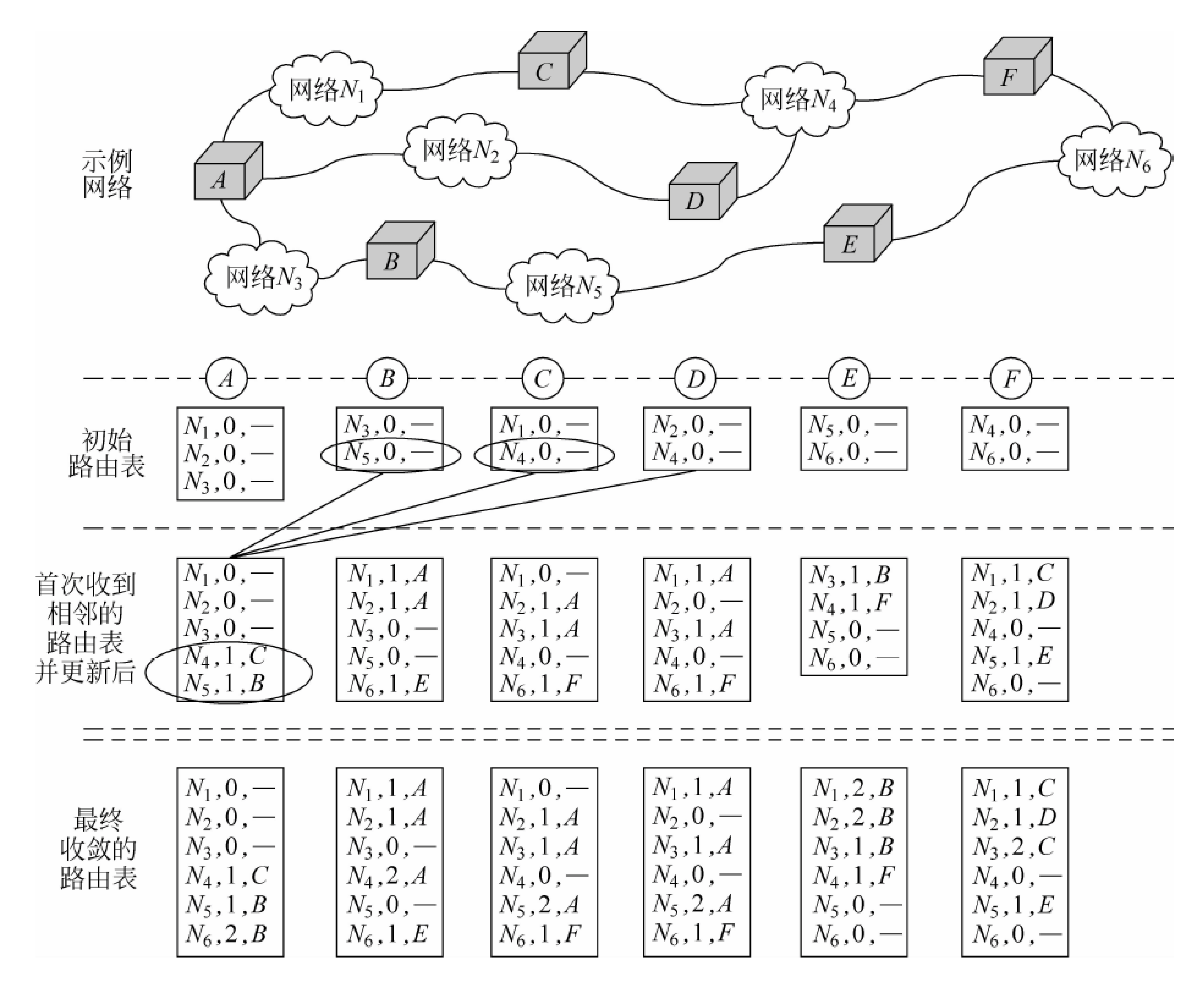


图 4.6　RIP 路由表更新示例

(1) 修改 RIP 报文中的所有记录项,将"下一结点"字段均改为 X_j,并将所有"距离"字段的值加 1(但不大于 16)。

(2) 对修改后的每个记录项,分别执行以下步骤:若"目的网络"不在路由表中,则添加该记录项,结束;否则若"下一结点"也相同,则替换原记录项,结束;否则若"距离"值小于路由表中的值,则进行更新,结束;其他情况不做处理,结束。

如果规定时间内(如 3min)未收到相邻结点的更新路由信息的 RIP 报文,则将此相邻结点记为"不可达"(距离值 16)。

需要注意的是,在图 4.6 所示的示例中,当第一次更新路由信息时,C 会分别收到来自 A 和 D 的路由信息,均包含可达网络 N_2 的记录项,且距离值都为 1,这种情况下,以先到达者为准,后到达者根据算法不做更新操作。虽然路由会因不同结点的 RIP 报文到达次序不同而不同,但是一旦确定后,将保持稳定。

另外,注意 RIP 算法第(2)步中,当接收到的记录项和已有路由表中"目的网络"和"下一结点"均相同时,应进行替换,而不论距离值孰大孰小,理由是可能因网络情况的变化造成路由改变,进而引起距离值的调整。如图 4.7 所示的实例说明了算法中这种替换的重要性。

如图 4.7 所示的例子说明 RIP 对网络故障可以做出正确反映,并修正路由表,但需要

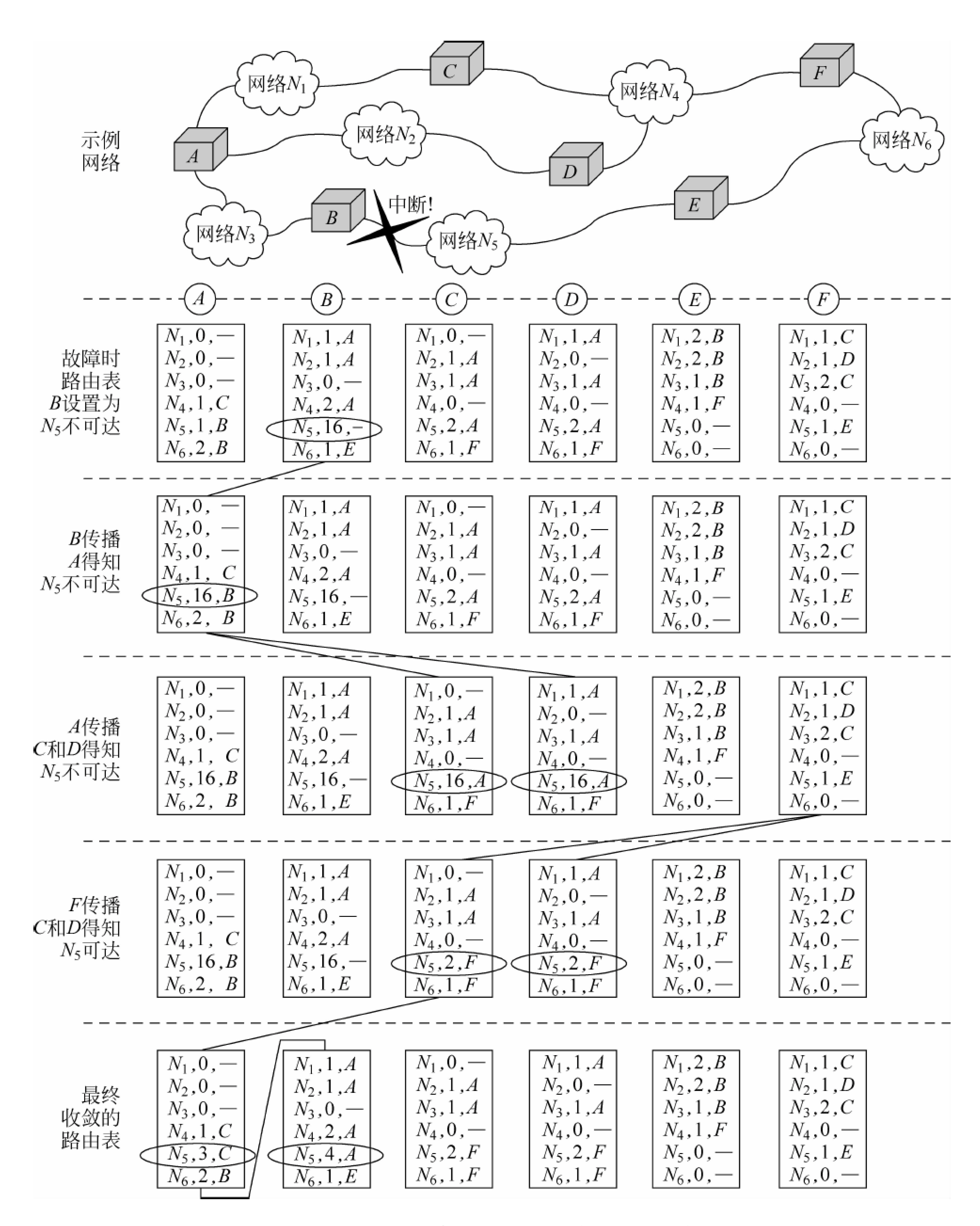


图 4.7　网络故障引起的 RIP 路由更新示例

一定的恢复时间。如图 4.8 所示的“极端”例子更证明了这一点。A 发现网络 N_1 故障，更新了路由表，由于 A 未到发布周期，B“抢先”发送路由更新，造成路由表来回刷新，必须经过多个更新周期直到距离值达到 16(不可达)后，双方才获得正确的路由表。更为严重的是，在这个较为“漫长”的收敛过程中，不但 N_1 网络故障的情况无法反映出来，而且引起了路由振荡。

RIP 采用 UDP(端口号为 520)进行传送，目前最新的版本为 RIP2(RFC 2453)。RIP2 的报文格式如图 4.9 所示。

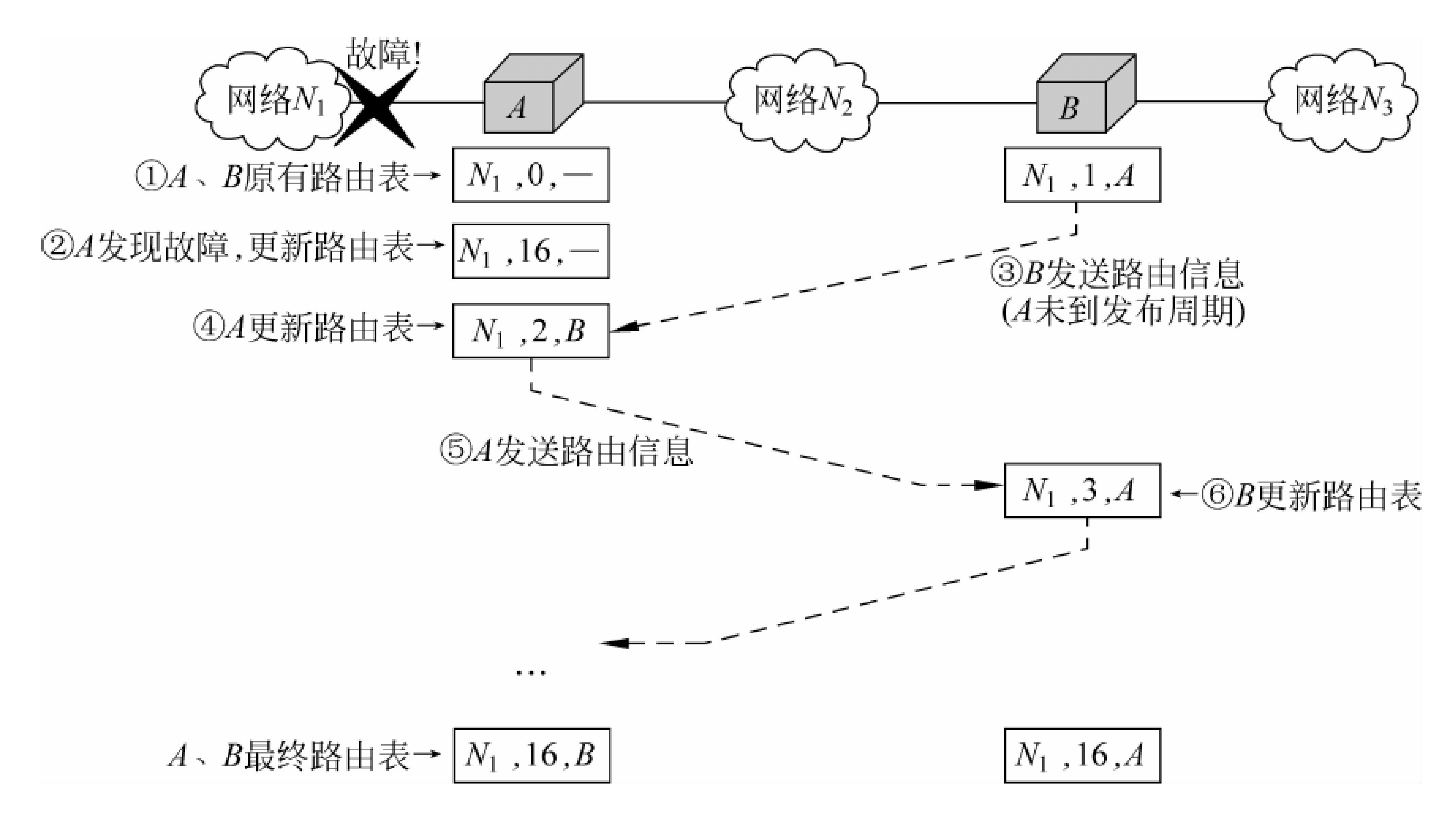


图 4.8 RIP 故障恢复的极端例子



图 4.9 RIP2 报文格式

RIP2 报文中,命令字段为 1 表示请求路由信息,2 表示对请求的响应或未被请求而发送的路由信息。

单纯以跳数作为选路的依据不能充分描述路径特征,可能导致所选的路径不是最优,因此 RIP 只适用于网络状况较为单纯和可预测的中小型的网络。毋庸置疑的是,RIP 简单而有效,所以能够成为路由信息交换的标准之一,几乎所有路由器均支持 RIP。

4.2.2 OSPF 协议

开放式最短路径优先(Open Shortest Path First,OSPF)协议是一种内部网关协议。目前采用版本 2,即 OSPF2(RFC 2328)。OSPF 采用分布式的链路状态协议(Link State Protocol),每台 OSPF 路由器都维护一个全网一致的链路状态数据库(Link State Database),实际上就是网络拓扑结构图,使用这个数据库可以构造一棵最短路径树来计算路由表。

OSPF 基于 Dijkstra 提出的**最短路径算法**(SPF),执行过程如下:

令 $D(v)$为源结点 a 到结点 v 的距离(包括从 a 到 v 的某一路径中所有链路距离之和),再令 $d(i,j)$为结点 i 到 j 之间的距离。

算法执行如下步骤。

(1) 令 N 表示网络结点的集合。先令 $N=\{a\}$。对所有不在 N 中的结点 v：

$$\begin{cases} D(v)=d(a,v), & \text{若 } v \text{ 与 } a \text{ 直接相连} \\ \infty, & \text{若 } v \text{ 与 } a \text{ 不直接相连} \end{cases}$$

(2) 寻找一个不在 N 中的结点 w，其 $D(w)$ 的值为最小。把 w 加入到 N 中。然后对所有不在 N 中的结点 v：

$$D(v)=\min[\,D(v),\,D(w)+d(w,\,v)\,]$$

(3) 重复步骤(2)，直到所有结点都在 N 中。

以图 4.10(a)为例，从结点 a 出发，按 SPF 算法逐步执行，如表 4.1 所示的各步骤和图 4.10(b)所示的结点扩展，遍历所有结点，最终得到图 4.10(c)所示的路由表。

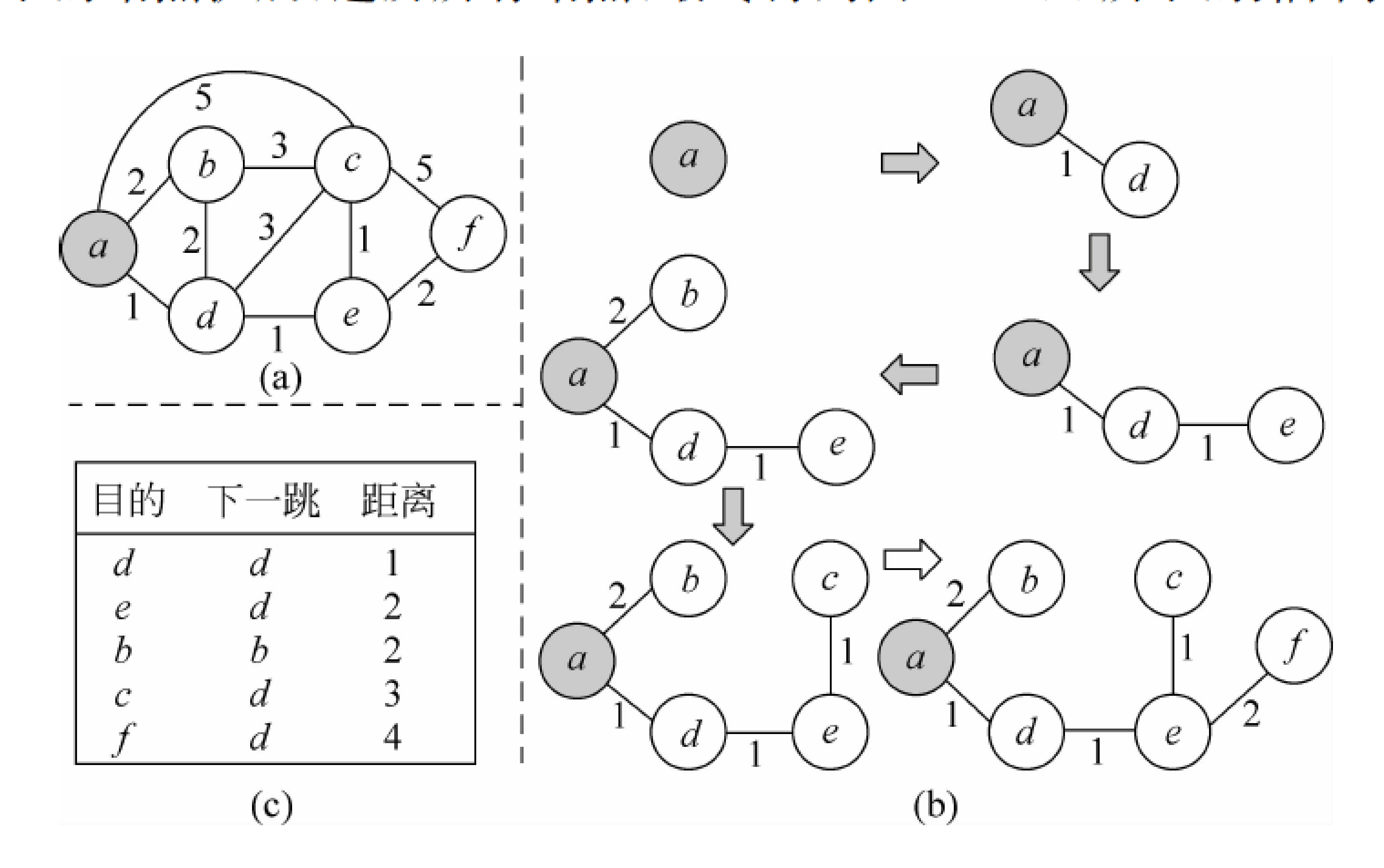


图 4.10　SPF 实例

表 4.1　SPF 算法分步计算表

步骤	N	$D(b)$	$D(c)$	$D(d)$	$D(e)$	$D(f)$
初始	$\{a\}$	2	5	1	∞	∞
1	$\{a,d\}$	2	4	(1)	2	∞
2	$\{a,d,e\}$	2	3		(2)	4
3	$\{a,b,d,e\}$	(2)	3			4
4	$\{a,b,c,d,e\}$		(3)			4
5	$\{a,b,c,d,e,f\}$					(4)

OSPF 路由信息采用洪泛法(flooding)发送：路由器向所有相邻的路由器发送信息，每台相邻路由器又向其所有相邻路由器转发此信息(不回发)。通过洪泛法，网络中的每台路由器都可收到一台路由器发出的同样的路由信息副本，且只收到一次。

如图 4.11，为了应用于大规模网络，OSPF 将一个自治系统划分为若干个区域(area)，分别分配 32b 的区域标识符(采用 IP 地址的点分十进制表示法)。为保障协议执行效率，一个区域内的路由器最好不超过 200 台。洪泛法仅在区域内运用，而非在全网范围，极大减少了网络上路由信息通信量。

因为一个区域内的路由器仅了解本区域的完整拓扑结构，不知道其他区域的网络拓扑情况，为实现区域间的通信，OSPF 采用层次化的区域划分。上层为主干区域(backbone

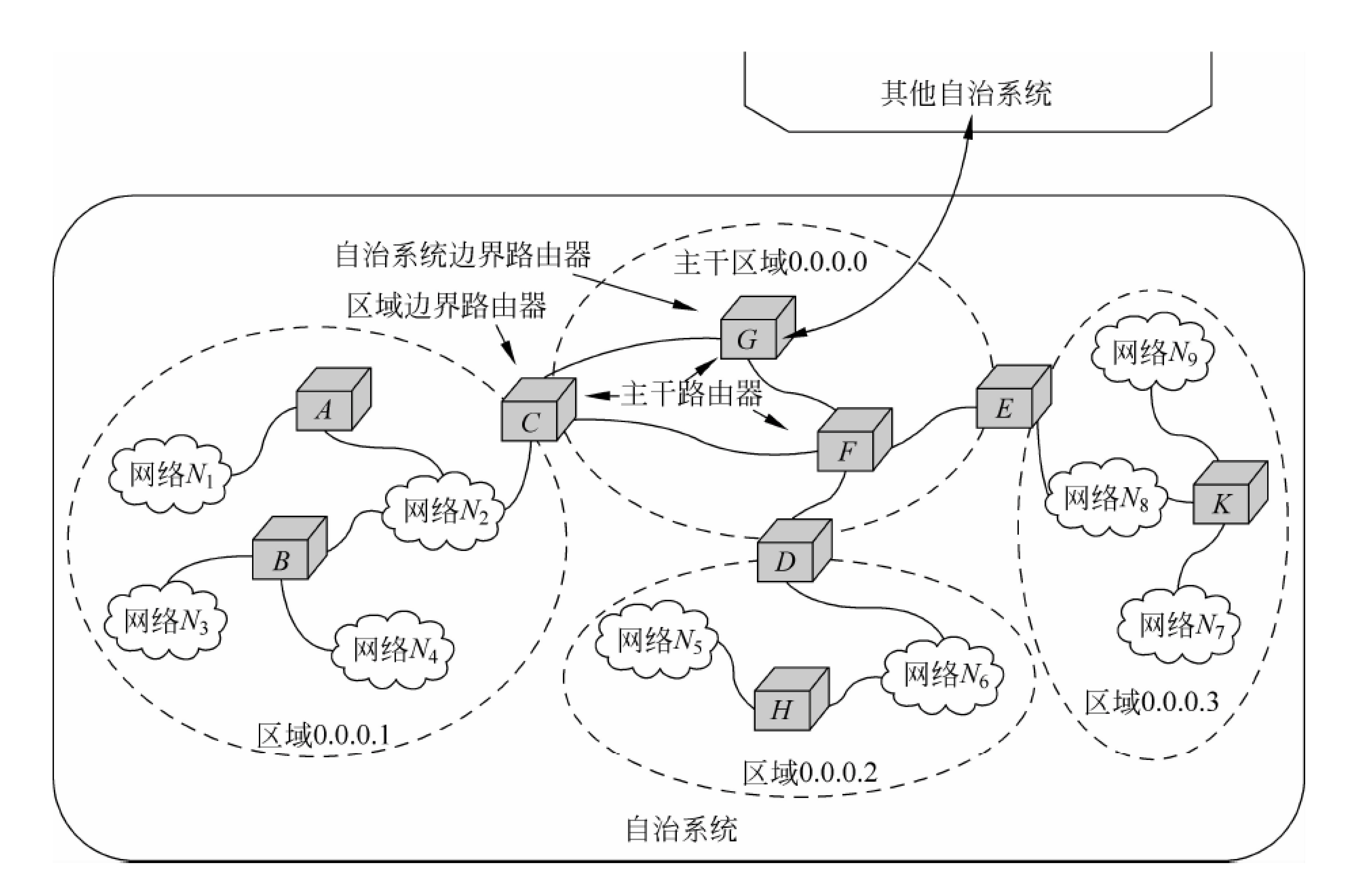


图 4.11　OSPF 区域划分示意

area),区域标识符规定为 0.0.0.0,通过区域边界路由器(area border router)连接其他下层区域,及通过自治系统边界路由器(AS border router)与其他 AS 通信。

OSPF 直接通过 IP 报文传送(IP 的协议字段值为 89)。OSPF 报文格式如图 4.12 所示。

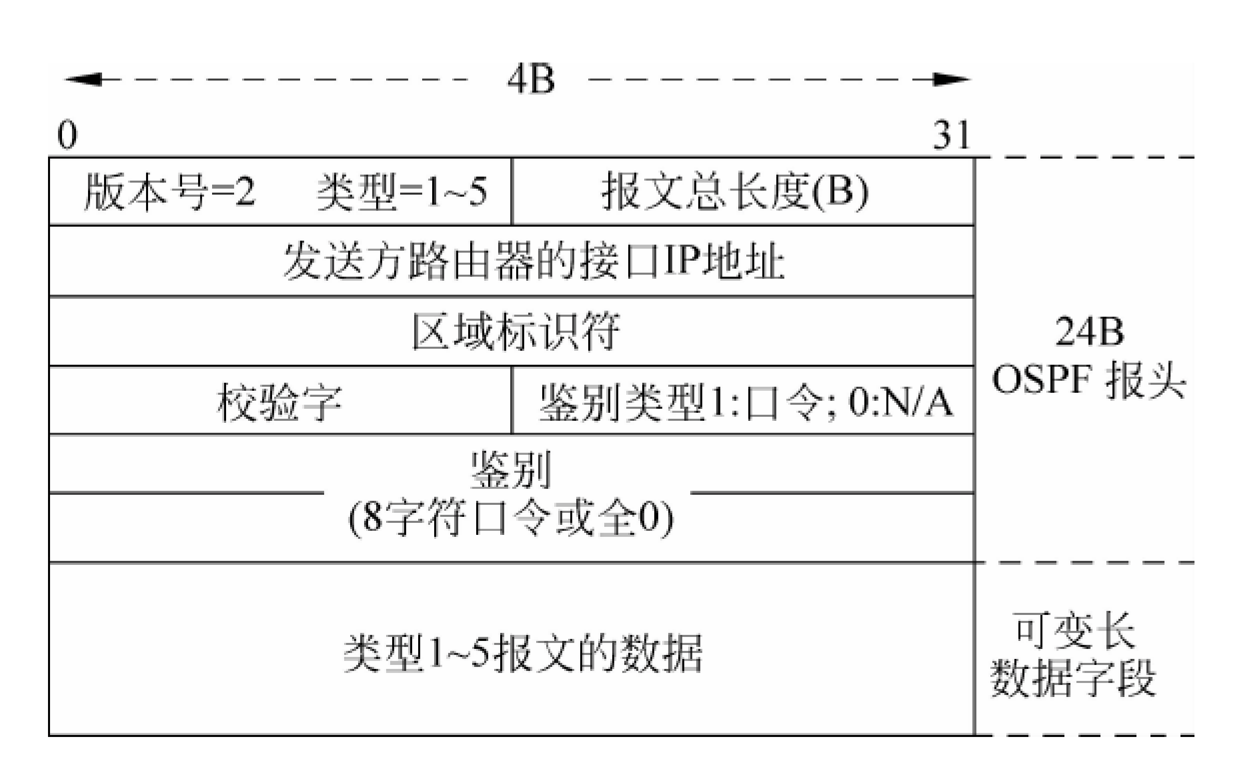


图 4.12　OSPF 报文格式

OSPF 有 5 种报文类型。

(1) 问候(Hello),发现和维持相邻结点的可达性。

(2) 数据库描述(Database Description),向相邻结点发送链路状态项目的摘要信息,即数据库中已有的结点的链路状态信息。

(3) 链路状态请求(Link State Request),向对方请求发送指定的链路状态项目的详细信息。

(4) 链路状态更新(Link State Update),用洪泛法对全网更新链路状态。具体又分为 5 种链路状态,讨论从略。

(5) 链路状态确认(Link State Acknowledgment),对链路状态更新的确认。

链路状态是指结点与哪些结点相邻,以及到达各相邻结点的距离、时延、带宽、费用等度量(或代价),均可由网络管理员来灵活设定。当且仅当链路状态发生变化时,才使用洪泛法发送更新路由信息。但是,如果所有结点都做本地链路状态信息的全网广播,则产生太大的开销。

按照 OSPF 规定,每两个相邻结点每隔 10s 要交换一次 Hello 报文,起到心跳信号(Heart-beat)的作用,确保结点是可达的,否则,若超过 40s 未收到 Hello,则认为结点不可达,应修改链路状态数据库,重新计算路由表。

在此基础上,结点采用数据库描述报文与相邻结点交换链路状态摘要信息,指出已有的信息(及其序号),然后,结点采用链路状态请求报文要求对方发送自己所缺少的某些链路状态项目的详细信息。

通过这一系列的信息交换,达到链路状态数据库的全网同步。

由于一个结点的链路状态只与相邻结点的连通状态有关,与网络规模没有直接关系,因此,OSPF 的收敛速度比 RIP 要快,而且在更新路由信息时产生的流量也较少。

如果把 OSPF 的所有距离矢量(或代价)都设为 1,则 OSPF 可以获得与 RIP 一致的路由结果。

OSPF 具有下列技术特点。

(1) 可根据 IP 报文的不同服务类型(ToS)设置不同的代价(1～65 535),由此计算出不同的路由。例如,高带宽的卫星链路,对文件传输业务可分配较低的代价,而对时延敏感的业务则可设定很高的代价。最为常用的是以链路带宽为代价。

(2) 如果到达同一个目的地有多条相同代价的路径,那么可以把通信量均匀分配给这几条路径,获得负载平衡(Load Balance)的效果。

(3) OSPF 报文是可鉴别的,保障了路由信息交换的安全性。

(4) OSPF 支持 VLS 和 CIDR。

(5) OSPF 为每一个链路状态带上 32b 序号,序号越大则状态越新,有利于协议机判别。不必担心序号溢出,因为序号增长周期要求大于 5s,可保证 600 年内不会重复。

4.2.3　BGP

边界网关协议(Border Gateway Protocol,BGP)是一种用于自治系统间的路由协议,主要功能是同其他的 BGP 系统交换网络可达信息,属于外部网关协议。最新为版本 4 的 BGP4(RFC 1771/1772)。BGP 使用 TCP 作为传输协议,端口号为 179。

在自治系统内部,网络主要通过运行 OSPF 和 RIP 等协议生成路由,而在自治系统之间,继续采用 OSPF 和 RIP 就完全不可行了。因为不同的 AS 具有各不相同的路由策略,无法采用一致的度量依据来构造穿越多个 AS 的最佳路径;其次,大规模的网络中情况十分复杂,要维持所有的链路状态非常困难,不仅路由表过于庞大、效率低下,而且更新时间必然相当长。这其实正是规划 AS 的原因之一,可以让网络的拥有者充分运用其管理权,自主治理 AS,而 AS 之间的路由问题交给骨干网络和 BGP 等协议来完成。

BGP 把每个 AS 看成一个整体,只负责为各个 AS 需要送往其他 AS 的数据报文指出有效的路径,使其通向正确的目的地。例如,互连各城域网,形成国家网,内部网络的相互访

问限制在国家 AS 内,不需要通过其他国家中转,再通过更高层次的互连网络,实现国家网之间的信息交换。

BGP 算法和协议遵循以下设计原则。

(1) 允许使用多种路由选择策略,灵活应对各种互连需求,综合考虑包括技术、经济、安全、政治等因素。

(2) 尽力寻找一条较好的路由,而非所谓的最佳路由。

(3) 避免循环路由(回路)。

(4) 以 CIDR 为基础,支持路由信息的聚合和削减。

因此,BGP 是以数据交换的可达性作为目标,采用路径矢量(path vector)计算路由,有别于 RIP 的距离矢量和 OSPF 的链路状态。

如图 4.13 所示,每个自治系统都要选择一台路由器作为 BGP 发言人(BGP speaker),运行 BGP,代表整个 AS 和其他 AS 交换路由信息。两个互连的 BGP 路由器彼此称为对方的邻居(neighbor)或称为对等体(peer)。对等体的连接有两种模式:IBGP(Internal BGP)和 EBGP(External BGP)。IBGP 是指单个 AS 内部的路由器之间的 BGP 连接,EBGP 是指 AS 之间的路由器建立 BGP 会话。

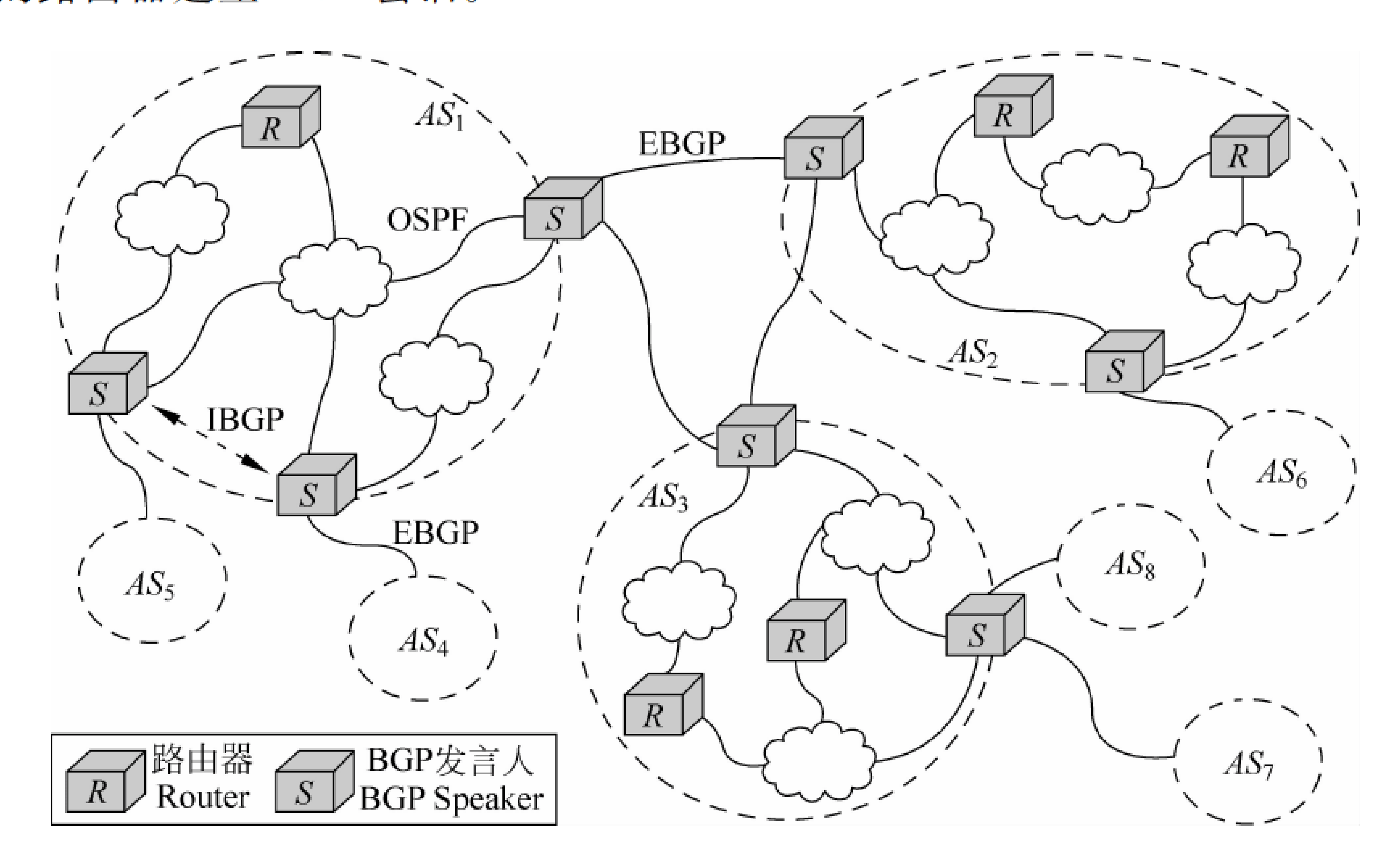


图 4.13　BGP 与 AS 互连示意

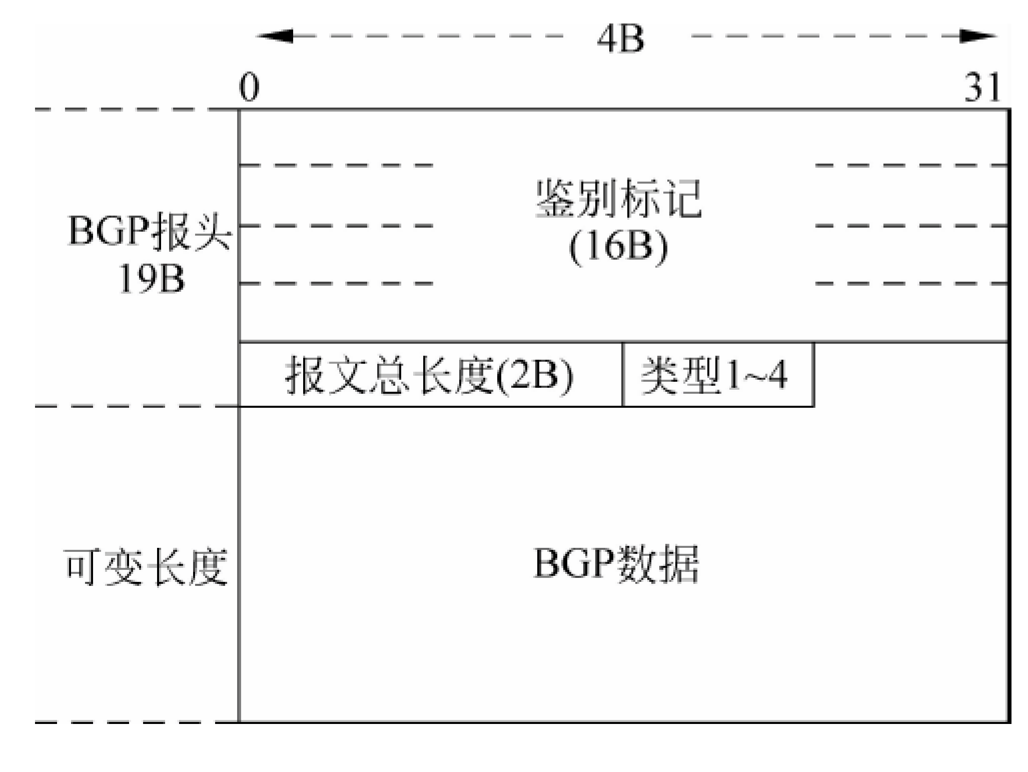


图 4.14　BGP 报文格式

BGP 有 4 种报文,报文格式如图 4.14 所示,具有 19B 的报头和可变长的数据字段。报头中 16B 的标记(marker)用来鉴别 BGP 报文,类似于密码(供扩展),如果不使用鉴别,则置为全 1。报文最大长度为 4096(包括报头)。类型字段编码为 1~4,分别表示 4 种 BGP 报文。

(1) 打开(Open)报文,类型=1。

用来与邻居发言人建立连接关系。路由器上线时,首先应当向所有邻居发送打开报

文，相当于一种声明及请求。如果邻居太忙或发生了故障，可以不做响应，协商失败；如果邻居同意接收，就响应保活报文，在这之后，就可进行路由信息的交换。打开报文包含 6 个字段(位于 BGP 数据字段中)。

① 版本号，1B，BGP4 为 4。

② 本 AS 编号，2B，全球唯一的 16b 编码。

③ 保持时间，2B，以秒(s)计，保持邻居关系的时间。

④ BGP 标识符，4B，一般为本路由器 IP 地址。

⑤ 可选参数长度，1B。

⑥ 可选参数。

(2) 更新(Update)报文，类型＝2。

用以发送某一路由的信息，并列出需要撤销的路由。更新报文就是 BGP 发言人之间交换路由信息的载体。在 BGP 刚开始运行时，需要与邻居交换整个路由表，以后在路由发生变化(增加、改变或取消)时才发送更新报文。更新报文有 5 个字段(位于 BGP 数据字段中)。

① 不可行路由总长度，2B，指明下一个字段(撤销的路由)的长度。

② 撤销的路由，列出所有需要撤销的路由。

③ 路径属性总长度，2B，指明下一个字段(路径属性)的长度。

④ 路径属性，定义在这个报文中增加的路径的属性，每次最多增加一条。

⑤ 网络层可达性信息(Network Layer Reachability Information，NLRI)，定义发出此报文的网络，包括网络前缀的比特数、IP 地址前缀。

(3) 保活(Keep-alive)报文，类型＝3。

仅 19B 的报头部分，没有数据部分。用以响应打开报文和更新报文，并且与邻居定时(如每隔 30s)交换，以保持活跃及维持邻居关系。

(4) 通知(Notification)报文，类型＝4。

发送检测到的差错信息，即故障报告。通知报文有 3 个字段。

① 差错代码，1B。

② 差错子代码，1B。

③ 差错数据，描述差错的诊断信息。

由于 BGP 路由器从所有邻居结点获得路由信息，了解各条路由的最新状态，便于选择路径。当某个路由器或某条链路发生故障时，BGP 路由器不仅可以及时获知，而且可以较为容易地选择一条新路径来替代。

CHAPTER 5

第5章 Ad-hoc 协议

自组网(Ad-hoc),或称**自组织网络**,是一种没有专门的基础设施(如AP)、无线结点直接进行相互通信的网络。在自组网中,无线结点之间具有对等性,可任意分布、可自由移动,因此具有拓扑结构的动态性、不确定性。由于结点的无线信号覆盖范围有限,结点间的通信可能需要其他结点中转,因此自组网又被称为**自组织多跳网**(Self-organized Multi-hop Network),而且自组网的所有结点的地位基本上是平等的,因此还被称为**无线对等网络**(Wireless Peer-to-Peer Network)。

Ad-hoc是拉丁语,意为仅为此目的(for this purpose only)、特定的。因此,Ad-hoc可理解为一种有特定目的的无线网络,其特殊性就在于可自行组网。

5.1 Ad-hoc 原理

自组网具有比较鲜明的特色,主要表现在:无中心、相互对等与自组织;多跳路由;动态拓扑结构;自动配置;有限的无线信道带宽;无线终端性能较低;能量有限;安全性较弱;网络技术的特殊性较强等。

自组网的无线结点既是终端,又是路由器,具有系统结构的独立性。结点一般由嵌入式处理机、无线通信模块、电源模块等组成,也可是PC、笔记本电脑、PDA或手机等。

实际上,传统的Internet虽然是以有线通信为主,但就是通过路由器结点相互协调来组网,而没有一个所谓的网络中心来集中控制,从这个意义上说,Internet何尝不是一个最大的Ad-hoc网络呢?所以,Internet的许多路由技术可以延用到无线Ad-hoc网络中。

根据骨干网络路由结点的移动性特点,自组网可分为**无线网状网**(Wireless Mesh Network,WMN)和**移动自组网**(Mobile Ad-hoc Network,MANET)两类。WMN骨干网络采用无线连接方式,路由结点移动性较小、性能较强,路由结点之间、路由结点和无线终端之间可使用不同的无线通信技术,一些路

由结点可具有有线网络接口（如接入 Internet）；MANET 的无线结点移动性较大，网络拓扑结构和连接关系与各个结点的位置相关，每个结点都需要为其他结点提供路由功能，结点间一般使用单一信道通信。

自组网通常有两种组网结构（如图 5.1 所示）：平面结构和分级结构。在平面结构中，所有结点都是平等的，是对等式结构；在分级结构中，结点被划分为**簇**（cluster），每个簇由一个簇头和多个簇成员组成，而簇头在逻辑上形成高一级的网络，并且还可以再分簇，构成更高一级网络。簇头负责簇间数据的转发工作。簇头可以预先指定，也可以由结点通过特定算法自动选举产生。

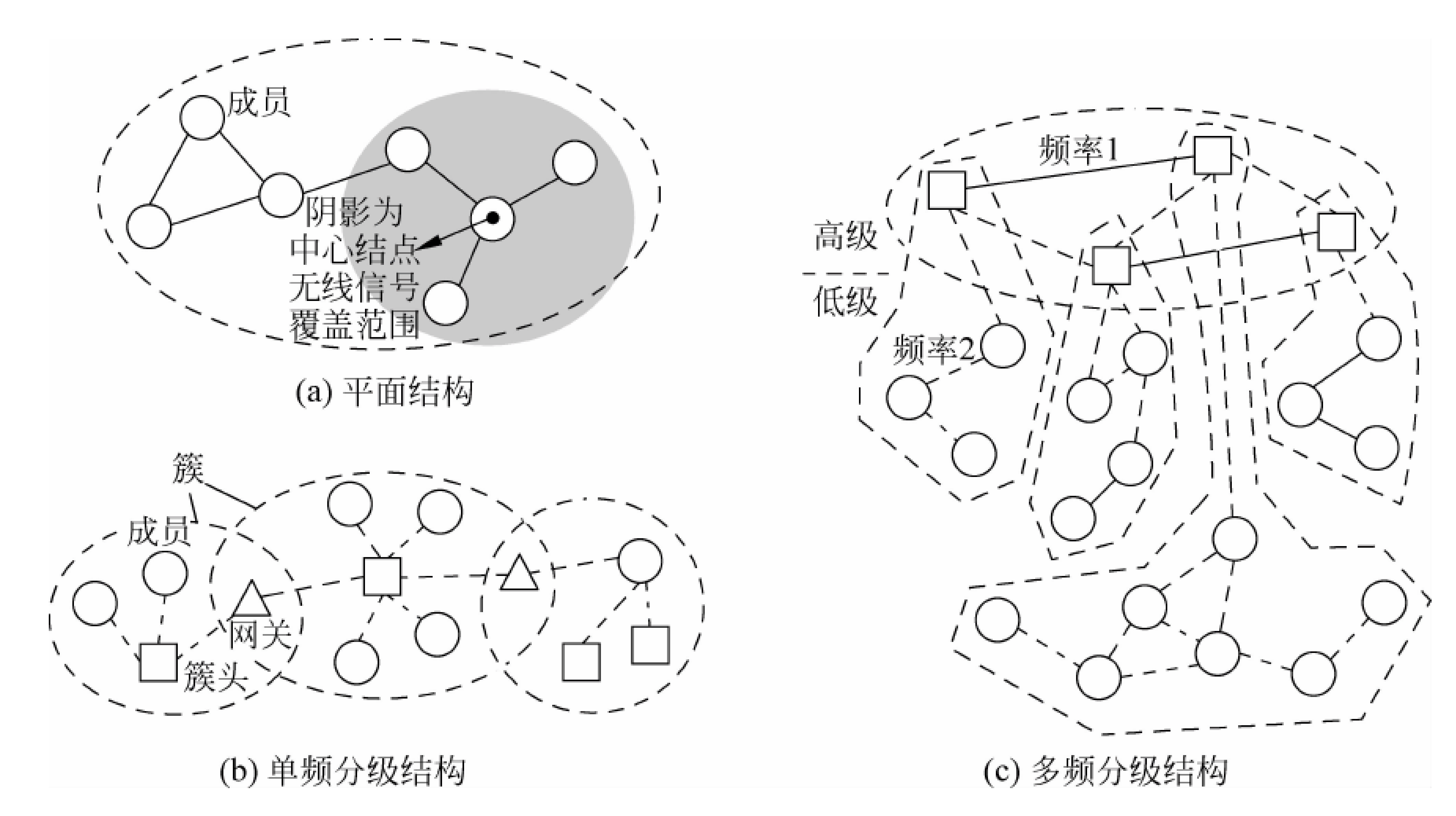


图 5.1　自组网结构示意

分级结构自组网又可进一步分为单频分级和多频分级两种类型。

自组网中需要注意**隐蔽站问题**（Hidden Station Problem）和**暴露站问题**（Exposed Station Problem），两者是可能引起数据接收冲突的潜在威胁因素。

如图 5.2(a)所示，当 A 发送报文给 B，由于 C 在 A 的覆盖范围之外，如果 C 此时也发送报文给 B 或 D，就会在 B 处产生碰撞，C 就是相对于 A 的隐蔽站。因为 C 作为发送者产生冲突，所以被称为**隐蔽发送站**。

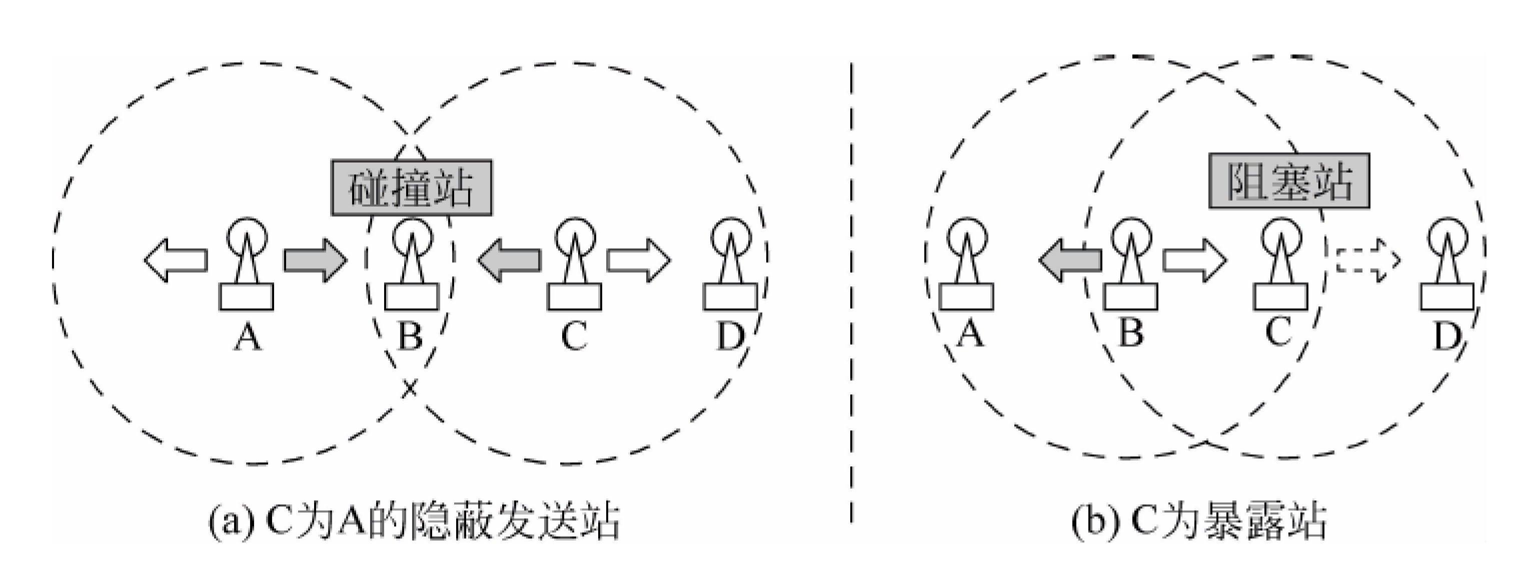


图 5.2　隐蔽站问题和暴露站问题

采用 RTS-CTS 握手机制可以解决隐蔽发送站问题，因为当 B 收到 A 发出的 RTS 而响应 CTS 后，C 同样可以收到 CTS，所以进入延迟发送状态。但是，如果此时 D 向 C 发送 RTS，C 无法做出响应，D 只能不断超时重发 RTS，浪费了资源，这种情况下的 C 称为 A 的

隐蔽接收站。因此握手机制解决了隐蔽发送站问题,却引发了隐蔽接收站问题,在单信道工作状态下无法两全其美。

使用双信道工作方式(注意区别于自组网多频分级结构),即数据和控制信息分别在不同的信道上传送,可以解决隐蔽接收站问题(试写出算法流程)。

如图5.2(b)所示,B向A发送报文,应当不影响C向D发送报文,但C在B的覆盖范围内,被阻塞而不能发送,则C就是暴露站。如果采用RTS-CTS握手机制,可在一定程度上解决暴露站问题:B向A发送RTS后,由于C收不到A向B回送的CTS,C可判断自己是暴露站,应可向D发送数据。

进一步分析,当C判断自己可以向D发送数据,就发出RTS,D于是回复CTS,然而,该CTS与B发送给A的数据报文在C处碰撞,使C收不到CTS,这样C就重复发送RTS,做无用功,这种情况下,C被称为**暴露发送站**。另外,若D要向暴露站C发送数据,发送的RTS同样会在C处冲突,C因此无法正常收到RTS,也就无法接收来自D的数据,这种情况下C被称为**暴露接收站**。这说明单信道通信实际上不能解决暴露站问题,易见采用双信道工作方式可以比较理想地解决两种暴露站问题(试写出算法流程)。

Ad-hoc协议主要有两类:一类是信道接入协议,即MAC层协议,这是系统运行的基础;另一类是路由协议,用于建立结点间的转发路径。路由协议有表驱动单播、按需单播、组播、广播、区域和分簇、相关性等不同种类。

MACA是用于Ad-hoc单频网络的信道接入协议,使用RTS-CTS握手机制,算法详见前述讨论,力求解决隐蔽站和暴露站问题,但不能完全避免。MACA是802.11WLAN标准的基础之一。除MACA外,典型的单信道接入协议还有IEEE 802.11 DCF(即CSMA/CA算法)、MACAW协议(MACA的改进)、FAMA协议(更好地解决MACA和MACAW中仍然存在的隐蔽站问题)等。

Ad-hoc路由协议设计的难点首先在于网络拓扑结构的不稳定性,所以需要快速适应动态变化,快速收敛。其次,Ad-hoc网络通常带宽较低、冲突较多,还应尽量节能,所以路由协议的高效性非常重要。此外,要防止路由干扰、路由窃取、路由劫持、黑洞攻击等安全威胁。Ad-hoc路由协议分为平面路由(包括表驱动、按需驱动)、层次路由、基于位置的路由等主要类型。

在设计Ad-hoc协议时,除了必须满足技术可行性、高效率、避免隐蔽站和暴露站等问题以外,还应注重降低能耗(包括单一结点的能耗和全局平衡能耗)、保障通信和路由安全、提供QoS支持、可管理等重要因素。

5.2 Ad-hoc路由协议

5.2.1 DSDV协议

目标序列距离向量(Destination Sequence Distance Vector,DSDV)属于表驱动平面路由协议,用于建立Ad-hoc网络的路由,实现结点间报文的多跳转发。每个结点维护一张路由表,记录该结点到网络中每一个目标结点的路由信息,属性包括目标结点、下一跳结点、到目标结点的跳数、由目标结点指定的序列号(就是目标序列名称的由来)。

DSDV 算法如下。

(1) DSDV 周期性地更新路由,或在拓扑结构发生变化时触发更新。更新路由报文包含内容有每个结点所能到达的目标结点、到目标结点的跳数、每条路径的序列号,即结点的路由信息表。

(2) 当结点接收到路由更新报文时,将跳数加 1,并与当前路由信息比较,旧的(较小的)序列号的路径被新的路由信息替代;如果序列号相同,就保存跳数较小的路径。更新路由后,需要将新的路由信息发布给相邻结点。

(3) 当发生链路中断时,通过该链路的所有路径的长度(跳数)被修改为无穷大,并更新路径的序列号(只有这种情况下序列号不是由目标结点指定)。

(4) 如果结点的一条路径长度(跳数)为无穷大,那么当接收到的路径序列号等于或新于当前序列号,将触发路由更新的广播。

例如在图 5.3 所示的拓扑结构中,结点 N_1 发生移动。以 $<N_d, N_n, H, S>$ 表示一条路由信息。当 N_7 发现与 N_1 的链路中断,就根据(3)将原有的到 N_1 的路由信息 $<N_1, N_1, 1, 175>$ 设置为 $<N_1, N_1, \infty, 176>$,并启动更新路由信息。其他结点陆续收到信息,根据(2),由于关于 N_1 的新路径的序列号较新,因此到 N_1 的跳数最终都等于 ∞,即不可达。

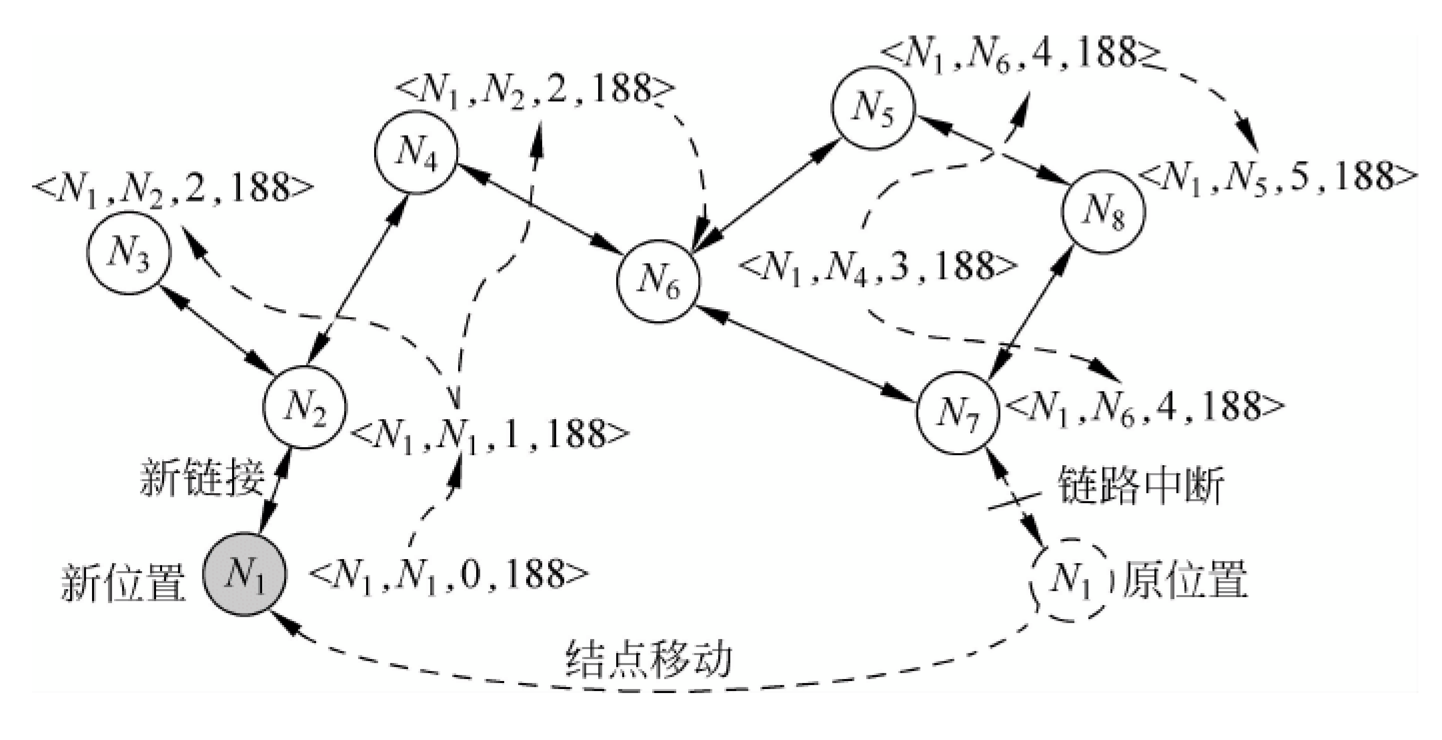


图 5.3　DSDV 路由更新示例

而当 N_1 到达新的位置后,向 N_2 发送 $<N_1, N_1, 0, 188>$,N_2 根据(2)更新路由表,并根据(4)广播路由信息,最终各结点将都获得图中所示的到 N_1 的收敛路由。可见 DSDV 类似 RIP。

DSDV 的优点是不产生路由环路,但难以确定最大稳定(收敛)时间、网络负载较重、不支持多路径路由。与 DSDV 同属表驱动类型的还有无线路由协议(Wireless Routing Protocol,WRP)、DBF(Distributed Bellman Ford)等。

5.2.2 DSR 协议

动态源路由(Dynamic Source Routing,DSR)属于按需路由的平面路由协议,是基于源结点的路由发现机制。

与表驱动路由协议的设计理念不同,按需路由协议认为在动态变化的 Ad-hoc 网络中,没有必要维护到所有结点的路由,只有在没有到达目标结点的路由的时候,才按需进行路由查找,当然数据报文会因建立路由产生延时。

按需路由协议由路由查找和路由维护两个过程组成。

如图5.4所示,源结点S广播到D的路由请求报文,相邻结点B收到路由请求报文后,记录报文经过了B,然后继续广播,请求从B到D的路由,直到到达D。结点D会收到多条来自不同路径的请求报文,每个报文中包含相应的路由信息,D根据一定的策略选择一条最优路径,并将该路由信息附在向源结点S发送的响应报文中,S收到响应报文后即获得了去往D的路由。当拓扑结构发生变化时,通过路由维护过程删除失效路由。

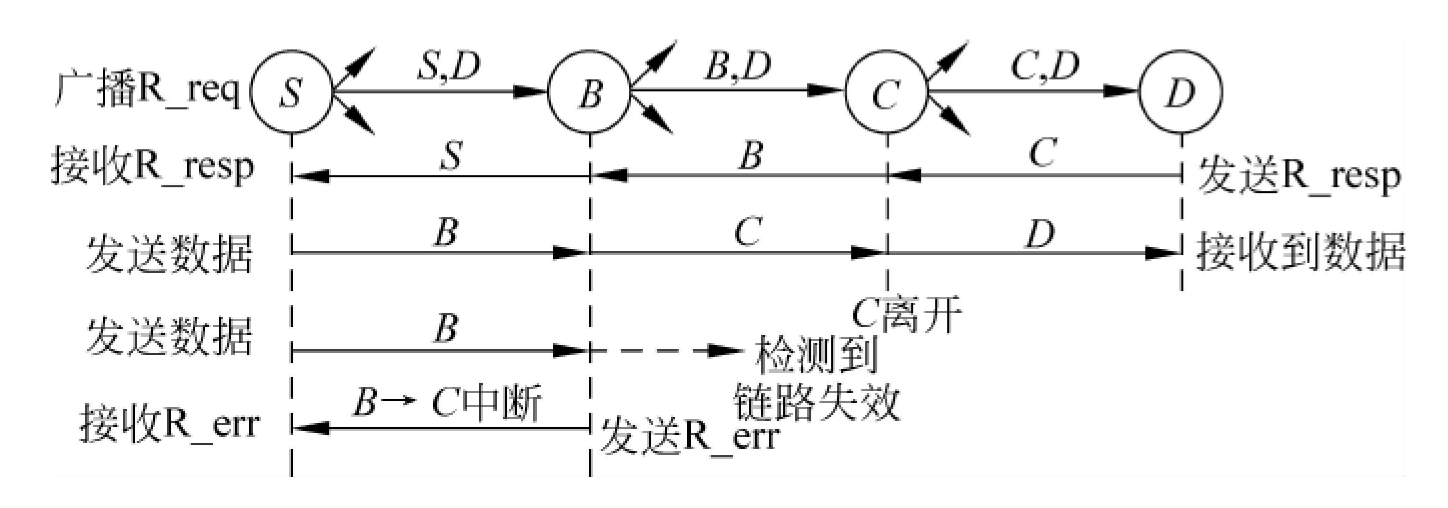


图5.4 按需路由协议基本流程

DSR的源结点在数据报头指定了到目标结点的完整路径,每个中间结点只需根据该路径将报文转发给下一跳结点。源结点首先检查路由缓存,若存在路径,则直接使用,否则执行路由查找协议。

DSR源结点发起的路由查找过程是一个优化的泛洪过程(即洪泛法),基本算法如下。

(1) 源结点广播发送R_req(路由请求)报文,包含唯一的ID号和初始值为空的列表,类似<S→D,ID,via{ }>。

(2) 当一个结点接收到R_req报文时,若已经见过该ID或列表中已经包含该结点,则到此终结,否则将自身添加到列表末尾,继续广播。

(3) 如果一个中间结点可以根据已有信息构建出从源结点到目标结点的完整路径,则将路径直接回复给源结点,不需要继续泛洪。

(4) 参与查找过程的结点都可以学习路径,并保存于自身缓存中。

(5) 如果收到两条以上的不同路径,则根据最短路径原则选择最优路径。

(6) 目标结点回复R_resp。

(7) 如果发现链路失效,结点将通过发送R_err报告路由错误,由源结点按需重新发起路由查找过程。

与DSR协议同属于按需单播路由协议的还有按需距离向量路由协议AODV(RFC 3561)、LAR、TORA等。

此外,基于组播的路由协议有:基于树的AMRoute、AMRIS,基于网状网(Mesh)的ODMRP、CAMP等;基于广播的路由协议有基于最小连通支配集的广播算法、竞争广播算法;还有区域路由协议(ZRP)、基于相关性路由协议(ABR)和单稳定性路由协议(SSR)等。

5.3 Ad-hoc网络

5.3.1 MANET

移动自组网(MANET)是Ad-hoc网络的主要类型之一,其特征是结点相互平等、一般

使用单一信道通信、结点移动性较强。

移动自组网不同于移动 IP(Mobile IP),虽然移动自组网通常使用 IP 为网络层协议。移动 IP 主要研究 Internet 的移动性用户终端和业务,包含有线通信链路在内,如漫游认证、自动配置、无缝切换等。

MANET 相当于一个实例化的 Ad-hoc 网络,除了运用链路接入、路由生成等技术实现结点互连外,还需要考虑以下方面的问题。

(1) 特定的应用系统设计。

(2) 数据链路层以上的协议栈规划、结点编址方法。

(3) 结点认证机制、数据加密机制、安全防范机制制定。

(4) 网络管理体系、管理域的建立。

此外,MANET 需要在应用系统相关的环境中对结点的移动建模,以优化技术设计、提高管理能力并提供决策参考。针对移动个体有随机移动模型、随机路点(Waypoint)移动模型、随机方向移动模型、城区移动模型;针对移动群组(相关个体)有队列移动模型、参考点移动模型、追逐移动模型;还有利用真实踪迹进行评估的基于踪迹的移动模型;等等。

在 MANET 设计中需要考虑对异常情况或边际效应做出正确处理,并防止错误蔓延,例如对入侵结点的预判和处理。入侵结点是指一个结点在通信过程中从一个网络(位置)移动到另一个网络(位置),与当地网络(位置)正在进行的通信产生冲突。

5.3.2　WMN

无线网状网(WMN)不同于其他无线网络之处在于它是一种基于多跳路由和 Ad-hoc 技术的网络结构,具有移动、宽带特性,可以动态扩展,比较适合大面积开放区域(如校园、小区、城市特定公共场所)的无线网络组网(如图 5.5 所示)。

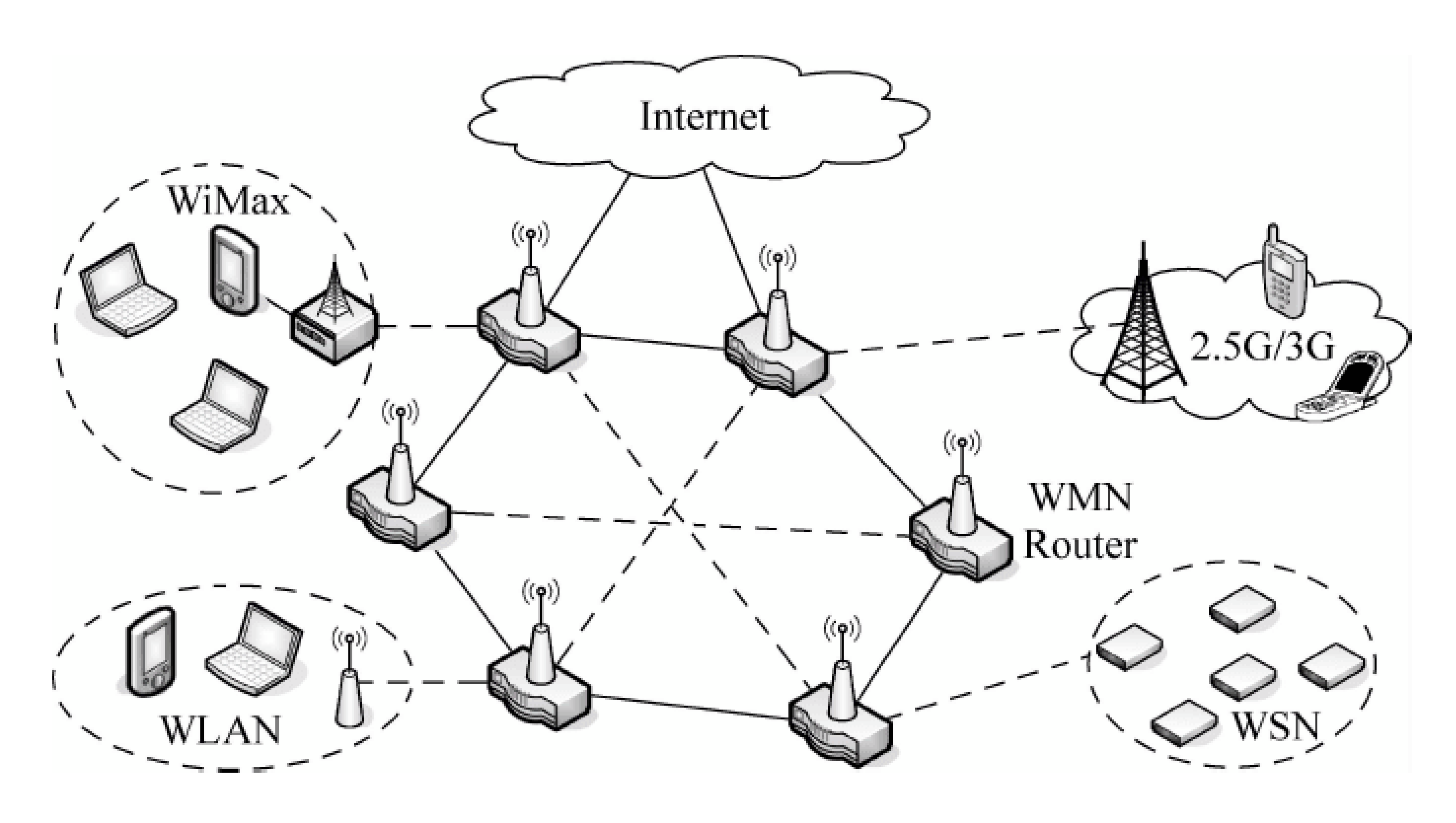


图 5.5　WMN 组网示例

WMN 为网状结构,所有结点间可相互连接、相互转发数据,一般选取部分相对固定的、性能较高的结点作为互连的路由结点。当两个结点间的一条链路失效后,WMN 路由器会经由一个或多个其他路由器寻找到一条替代路径。普通结点也可以通过结点接力绕开障碍物,使用最佳的传输路径,体现了网状网所应有的健壮性。由于一条路径可以由多个结点构成,每个结点只需与相邻结点通信,从而可以降低总功耗。实际上,Internet 就是 Mesh 网络最典型的例子(有线 Mesh 网络),任意两个结点间的通信都可以通过其他多个结点(路由

器)进行转发(多跳)。

在 WMN 组网中,可以充分运用 WLAN、WiMax、MANET、WSN 等各种技术,提供面向多媒体业务、便捷接入、高速通信、自由移动的服务。

为提高无线通信频率的使用效率,并尽量减少相互干扰,WMN 常采用智能定向天线技术,成为其技术特色之一。智能定向天线是一种信号功率集中的指定方向波束成形技术,采用**多输入多输出**(Multi-Input Multi-Output,MIMO)机制,使用相位受控的 m 个天线振子组合,可形成 m 个不同方向的低功率定向发射,使到达接收点的信号功率最强,而对其他邻近站点的辐射最小、影响最小,实现网络密集覆盖的低功率应用。在不能直接利用智能天线的场合,也可以采用 MIMO 技术,以提高功率效率和传输效率。

在 WMN 中可能存在一些特殊的网络,例如一种接入 WMN 的 Ad-hoc 网络,允许有输入或输出流量,但不允许有途经(穿越)的流量,这种网络被称为**残桩网络**(Stub Network)。

基于 WMN 技术可以设计各种有效的应用,例如:

(1) 园区无线网络。利用 WMN 技术可以使用较少的网络资源、投入较低的成本组建园区网络,并接入 Internet,开展特色应用服务。

(2) 突发事件(救灾)现场应急指挥网络。WMN 可以提供移动、宽带和灵活的自组网通信,在重大事件或重要活动的现场,能够迅速建立无线网络,实现现场指挥人员和处警人员之间的数据、语音、实时视频通信,并能够对处警人员精确定位,将现场的画面和数据实时回传给指挥中心,以此作为现场决策的重要依据。

(3) 智能交通管理。先进的智能交通管理系统,包括交通信号控制系统、交通视频监控系统、交警车辆调度、交通信息提示牌、车辆定位等功能,需要依靠移动宽带网络的支撑,WMN 正是解决方案之一。

(4) 大型赛事组网。在任何时间、任何地点享受各种形式的信息服务已经成为一项基本需求。WMN 可以满足人员临时性强、密集度大、流动性快、移动性高的特点,为各种类型的终端提供高速接入服务。

5.3.3 WSN

1. WSN 原理

无线传感器网络(Wireless Sensor Network,WSN)是带有传感器的微型系统组成的 Ad-hoc 网络。由于 WSN 结点体积小、分布广,也被形象地称为智能尘埃(Smart Dust)。

WSN 体现了无处不在的信息采集方法,是普适计算(Pervasive Computing)的重要基础性技术。如图 5.6 所示,WSN 可随机投放到目标环境中,借助本身携带的各种类型传感器,测量区域内的温度、湿度、噪声、光强、气压、震动、移动,以及电磁波、红外线、放射性、土壤成分、运动方向和速度、物体大小等,并将采集到的信息通过 Ad-hoc 联网、汇集并传递到管理中心(观察者)。

WSN 的应用将会非常广泛。在军事上,WSN 可用于敌情侦察、战场遥测、预警栅栏、核武器或化学武器爆炸现场勘察等;在环保方面,可用于气象参数采集、环境状况监测、地震和水灾预报等;在城市管理方面,可用于交通流量监测、物流调度、定位服务、不停车收费(Electronic Toll Collection,ETC)等;在日常生活中,可用于安保监控、家居环境、健康指标测量、宠物管理等。

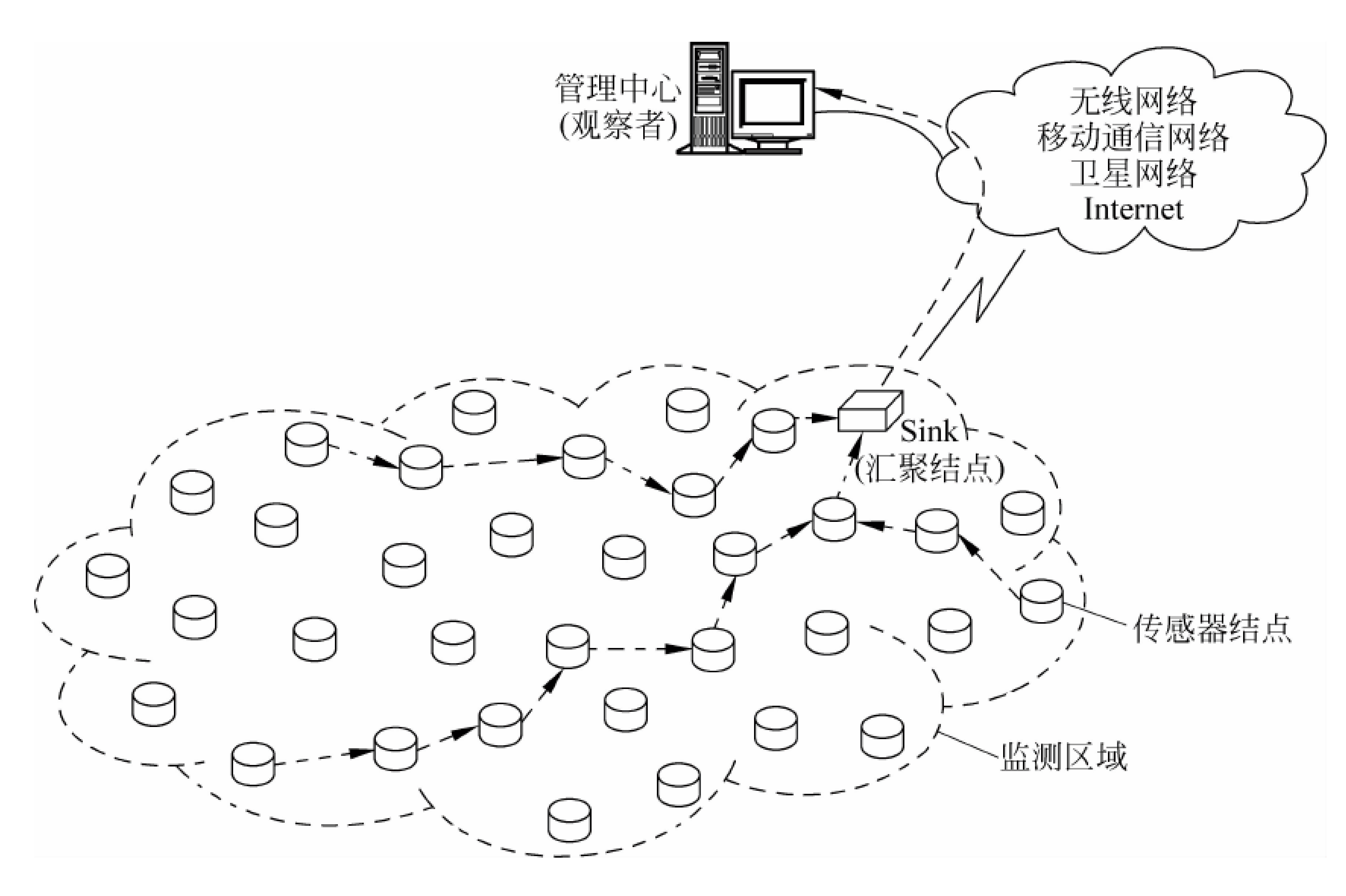


图 5.6　WSN 系统示意

由此可以看出 WSN 和其他 Ad-hoc 网络的差别：WSN 的明确目的是通过各个结点获取信息，网络中信息传递有一致的方向，所有结点要么是信息源，要么是中转站，而不可能是信息传递的终点；而且，一旦 WSN 部署完毕，其结点通常不具有移动性，或移动性不强，因此拓扑结构相对稳定。

然而，正因为 WSN 结点需要独立工作，且工作环境往往十分恶劣，所以其可靠性、适应性、容错性、相互协作能力非常重要。其中能量供给（电池）是最薄弱的环节，能耗决定了结点的工作寿命，而且可能影响到局部或全局的网络。尤其是因为信息传递总是指向管理中心方向，越靠近管理中心的结点通信任务越繁忙，必然造成能耗较大，容易缩短生命周期，而一旦这些结点受损，管理中心将永久性失去与 WSN 的联系。另外，WSN 在军事应用中一般位于敌对方的区域，因此安全保障、自我防范能力也必须强调。

WSN 的结点主要有两种类型。

（1）**传感器结点**（Sensor Node）。传感器结点是数量最多也是最重要的结点。网络中的部分传感器结点还承担中继转发任务，可称为转发结点、中间结点、过渡结点或中继结点，转发结点可由专门从事数据转发的结点来担任。如图 5.7 所示，传感器结点主要由处理单元、传感单元、通信单元、能量单元等组成，一些特殊的结点还可包括定位系统和移动（运动）系统。

（2）**汇聚结点**（Sink）。汇聚结点在 WSN 中具有唯一性，承担信息的汇集并通过远程通信手段传递给管理中心。汇聚结点一般有较高的性能，但也容易成为系统的单点故障点。

WSN 的体系结构由分层的网络通信协议、网络管理平台和应用支撑平台三部分组成，如图 5.8 所示，每个部分由不同的层次或模块（子系统）构成。

传感器结点对目标区域的有效覆盖状况直接影响到 WSN 所能提供的感知质量，应当针对被监测对象及其监测区域的情况，对传感器结点进行合理部署。主要有三种覆盖类型。

（1）区域覆盖：目标区域内的每一个点（每个对象）至少被一个结点所覆盖，同时保证

网络内各结点之间的通信连通性。这种方法往往容易造成覆盖冗余。可将高密度部署的结点划分成若干个互不相交的结点集合,每个结点集合都能维持对目标区域的原始覆盖质量,通过轮换各个集合的工作时间以节省能耗,延长整个系统的寿命。

(2) 点位覆盖:要求对目标区域内的有限个离散点进行监测,需要确定覆盖这些点的最少结点数及结点位置,并保证结点之间的连通性。可采用构造覆盖点集的最小生成树等方法。

(3) 栅栏覆盖:用于监测某个移动目标是否到达或正在穿越某个区域,由于结点的感知能力随距离增大呈指数型衰减,应通过对目标的危险程度、移动方向、移动速度等进行预估,合理部署结点,使目标穿越网络时被检测到的概率最大。

WSN 数据链路层 MAC 协议最为关注能耗的最小化,为此可以适当牺牲带宽、吞吐量、时延等 QoS 性能。

WSN 链路接入协议有固定分配、竞争占用两类。前者有 FDMA、TDMA、CDMA 等三种,如 DEANA、LMAC、TRAMA、DMAC、M-ALOHA、M-PSMA 等;后者一般在广播式信道上工作,在信号交叉范围内各结点共享信道资源,如 S-MAC、T-MAC、WiseMAC、Sift 等协议。

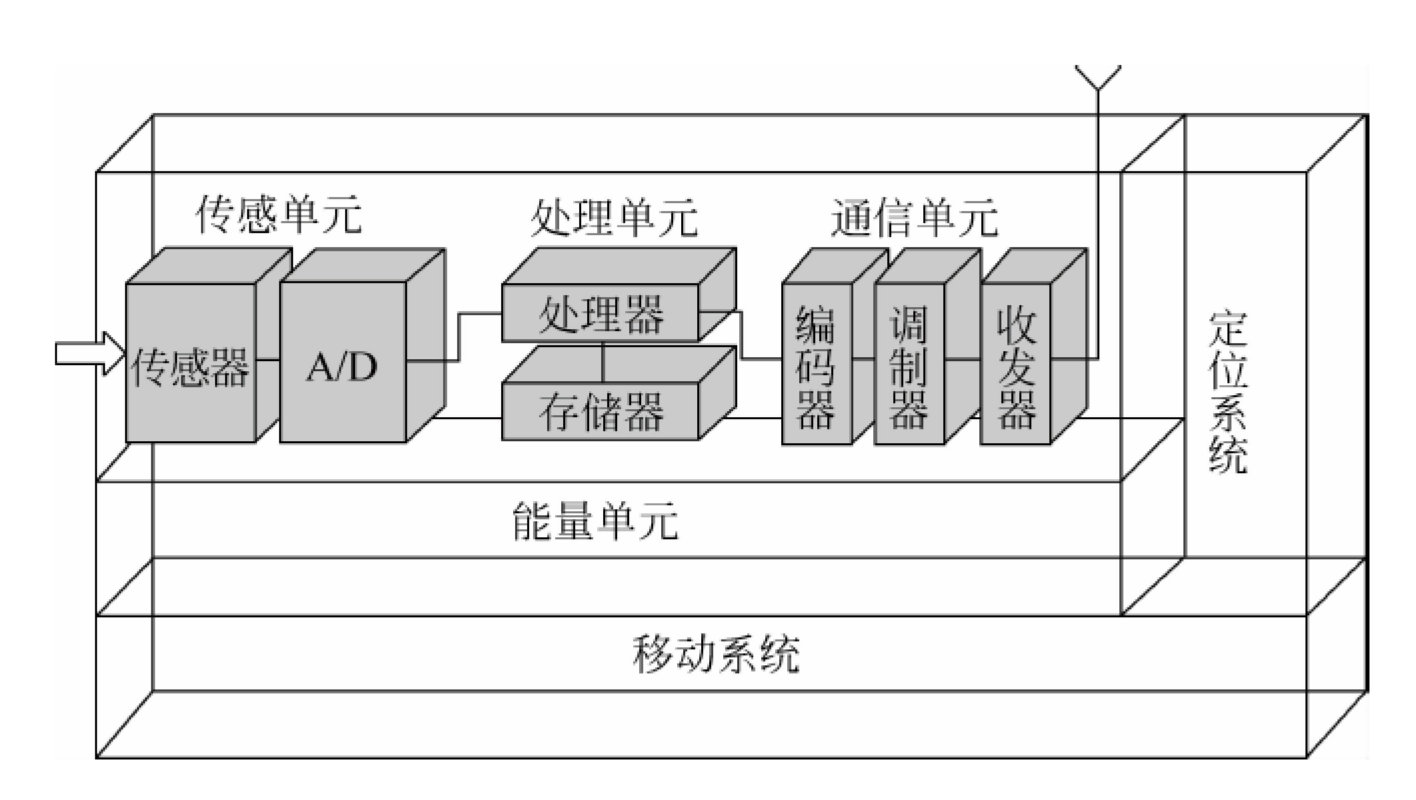


图 5.7 WSN 传感器结点系统结构示意

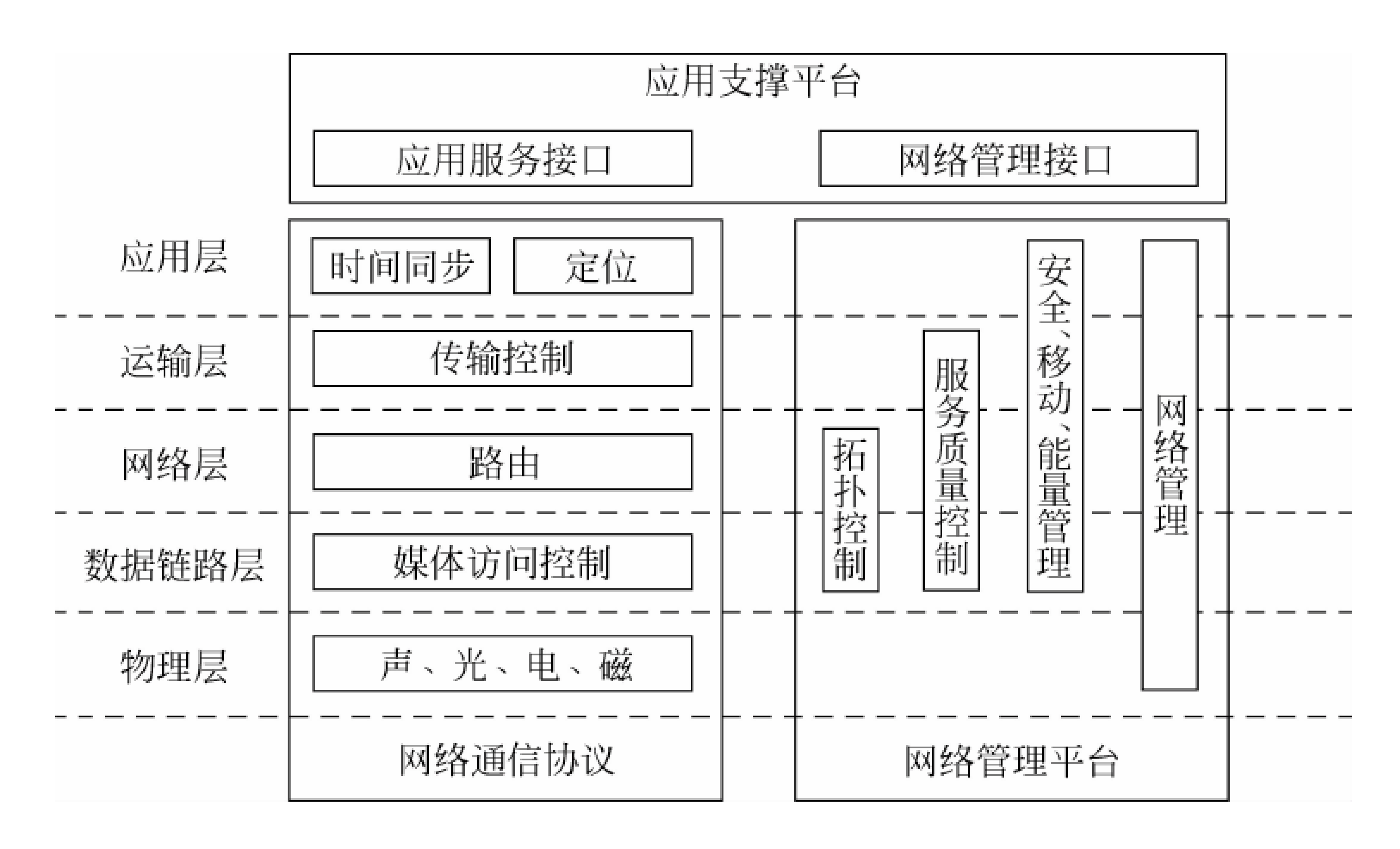


图 5.8 WSN 体系结构

(1) TDM-FDM 混合协议。由 FDMA 将频带分为多个信道，不同结点可以同时占用不同的信道(并发工作)；TDMA 则将一个时段(时隙)内的频带资源分配给一个结点使用。由于信道和时隙需要预先规定，可能限制结点对空闲时隙的有效利用，降低了信道利用率，而且 TDMA 需要时间同步开销，增加了能耗和系统复杂性。

(2) S-MAC 链路协议。S-MAC 协议考虑了 WSN 的三种状态：休眠态、空闲态和活动态，有针对性地采取相关策略以降低能耗，如结点定期睡眠、减少空闲监听；邻近结点组成虚拟簇，自动同步睡眠调度时间；用消息传递的方法减少时延。但休眠机制的设计较为困难，容易造成 over emitting 现象，即接收端处于休眠或唤醒后尚未准备好，造成发送端的能量浪费。S-MAC 采用类似 IEEE 802.11 的 CSMA/CA 方法来避免冲突，包括物理载波侦听、虚拟载波侦听和 RTS-CTS 握手机制，根据流量情况，在能耗和性能间取得较好的平衡。

2. WSN 路由协议

与 MAC 协议类似，WSN 的路由协议除了需要构建优化的路由，还应该着眼于减少能耗、达到负载平衡，因为一个结点的能量耗尽将意味着拓扑结构的改变，甚至还会引起网络被割裂成几个部分，使情况更为恶化。

WSN 路由协议算法很多，根据技术思路大致可分为 5 类：基于洪泛的路由协议、基于位置(地理)的路由协议、分簇路由协议、地理分簇路由协议和以数据为中心的路由协议。

(1) SPIN(Sensor Protocol for Information via Negotiation)协议。采用 SPIN 协议的结点以广播(洪泛)方式发送路由信息，接收消息的结点再以广播(洪泛)方式转发消息，虽然与拓扑结构无关，且控制机制简单，但洪泛法容易引起严重的冗余、冲突及广播风暴。一种改进的方式是 GOSSIPING 方法，结点发送数据时不采用广播，而是随机选择一个相邻结点。

SPIN 协议采用受控的洪泛机制，引入元数据(meta data)即报文的消息摘要，结点先广播元数据，当遇到相应的数据请求，才有目的地发送完整数据报文。ADV 消息用于宣布有数据要发送(包含元数据)，REQ 消息用于请求希望接收到的数据，DATA 消息用于封装数据。

(2) GPSR(Greedy Perimeter Stateless Routing)协议。GPSR 协议将报文的转发方式分为两种模式：贪婪(greedy)模式和边缘(perimeter)模式。在贪婪模式中，结点将报文转发给地理上更靠近目标结点的邻居结点，因此属于基于位置的路由。当贪婪机制失败时，就进入边缘模式，结点按照“右手法则”选择下一跳结点，就像迷宫游戏中的“右手贴墙”秘诀，保证了报文的正确转发。

GEAR(Geographical and Energy Aware Routing)协议是对 GPSR 的改进，综合考虑距离与剩余能量的关系，避免导致各结点的负载不均衡性。GEAR 同样先采用贪婪模式，失败时选择代价较小的邻居结点作为下一跳。如果迭代使用 GEAR 协议(划分子区域)，还能够建立 Sink 结点到检测区域多个数据源结点的节能路径。

(3) LEACH(Low-Energy Adaptive Clustering Hierarchy)协议。LEACH 属于分簇层次型路由协议。网络被动态划分为若干个簇，簇内结点采用最短路径算法将数据发送到簇头，由簇头对本簇成员的数据进行综合处理，再发送给 Sink 结点。

LEACH 算法按轮次来选择簇头结点，每轮分为设置(setup)和稳定(steady)两阶段。在设置阶段，网络随机选出部分结点作为簇头，簇头进行广播，普通结点根据接收信号的强

弱来投靠最近的簇头,与之共同形成一个簇;在稳定阶段,簇成员把收集的数据传给簇头,簇头将收到的数据与自身数据进行聚合处理,通过一跳的方式将结果发送给Sink结点。由于每一轮都重新随机选取簇头,使能耗可以均匀地分布到所有结点上。

TEEN是LEACH的改进算法,用于事件驱动型的WSN。TEEN定义了硬和软两个门限值,以确定是否需要发送监测数据。当检测数据超过硬门限时,就发送数据,并将该监测数据设为新的硬门限值;在接下来的过程中,如果监测数据的变化幅度大于软门限规定的范围,结点发送数据,并将其设定为新的硬门限值。通过调节软门限值的大小可以在监测精度和能耗间取得平衡。

PEGASIS是进一步改进的算法,可以减少簇头结点数量。基本思想是:把系统中所有结点用贪婪算法构成一个边长之和接近最小的链,然后随机选择一个结点作为簇头;簇头向链的两端(两个方向)发出收集数据请求,数据从链的两个端点向簇头流动;中间结点将接收的数据与自身数据进行聚合处理后,把结果发送到相邻结点,直到到达簇头,由簇头将最终结果发给Sink结点。为了减小单链引起的时延,可使用多链结构。

(4) GAF(Geographical Adaptive Fidelity)协议。GAF是一种典型的地理分簇路由算法,综合了基于位置的路由和分簇路由的优点。GAF把监测区域划分为多个虚拟的单元格,在每个单元格中定期选举产生一个簇头,簇头保持活跃状态,其他结点进入休眠状态。

GAF虚拟单元格的边长 r 满足 $r \leqslant R/\sqrt{5}$,R 为结点的通信半径,使相邻单元格中的任意两个结点都能够直接通信(数学原理如图5.9所示)。

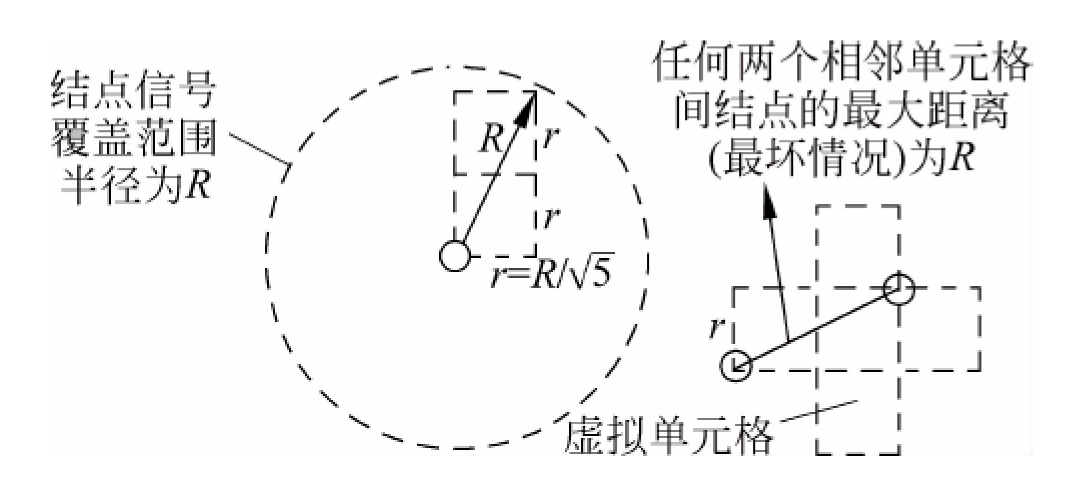


图5.9 GAF算法虚拟单元格

更“严酷”的情况是考虑对角线位置相邻(假定十字相邻位置的结点都已失去),这时两个结点间通信的最远距离应为 $2r\sqrt{2}$,则需要满足 $r \leqslant R/\sqrt{8}$。

假定所有结点的通信半径 R 均相同,并且所有结点都已知整个监测区域和自身所处的位置,那么容易计算自身所处的单元格。

GAF定义簇头选举按以下策略进行:结点周期性地进入休眠(sleep)状态或工作(active)状态;休眠结束或工作结束进入一个短暂的发现(discovery)状态;结点进入发现状态时,通过报文交换来选择簇头;选出的簇头进入工作状态,其他结点进入休眠状态。

(5) 定向扩散(Directed Diffusion)协议。定向扩散协议属于以数据为中心的Ad-hoc路由协议。其算法可描述为以下三个执行阶段。

① 路径建立阶段。Sink结点用一组称为兴趣(interest)的属性描述其查询请求,并向全网广播,逐级扩散,最终遍历全网,寻找所有匹配的原始数据,同时建立一个称为梯度(grad)的度量与兴趣发布过程相关联。

② 数据传播阶段。如果数据源结点的数据与兴趣相匹配,则按照兴趣传播的反向路

径，以较低速率将测试数据发送给 Sink。

③ 路径增强阶段。Sink 接收到的测试数据可能经过多条不同路径，可按照一定策略（如最低时延、最短路径等）选择一条路径进行路径增强，作为后续数据传输路径。

在多数据源情况下，数据转发存在交汇的中间结点，则由交汇结点执行数据聚合操作，以减少数据传输量并降低能耗。虽然定向扩散协议仅采用"随遇而聚"思想，没有对数据聚合进行优化，但对后续的改进工作有很大启发。

由于中间结点的数据聚合，相当于在 Sink 和数据源结点间构成了一棵组播路由树，因此可以把 WSN 的最优聚合问题转化为最优聚合路由树问题。

CNS、SPT、GIT 是定向扩散协议的三种变化。

CNS 方法选择距离 Sink 最近的一些数据源结点为聚合点，其他数据源都把数据发送给这些聚合点进行聚合处理。

SPT 方法要求每个数据源都分别沿着最短路径传输数据给 Sink，如果这些最短路径发生交汇，则交汇结点就是聚合点。

GIT 方法采用逐步建立聚合树的办法，树的初始化主干为 Sink 到距离最近的数据源结点（保证是最短路径），此后的每一步中，从剩下的数据源结点中选出距离路由树最近的那个结点连接到树上，直到所有结点都连接到树上。

（6）谣传路由（Rumor Routing）协议。谣传路由协议是对定向扩散路由协议的改进，用于减小洪泛传播和路径增强机制引入的开销，适用于数据传输量较小的 WSN。

谣传路由协议采用查询消息的单播随机转发方法，在事件区域中的结点产生代理（agent）消息，代理消息沿随机路径向外扩散传播（就像一则口口相传的谣言），同时 Sink 发送的查询消息也沿着随机路径在网络中传播，当两条路径交叉在一起，就会形成一条从 Sink 到事件区域的完整路径，于是交叉结点会沿着查询消息的反向路径将事件信息传递到 Sink 结点，如图 5.10 所示。但是谣传路径不是最优路径，还可能存在路由环路问题（由于随机选择造成）。

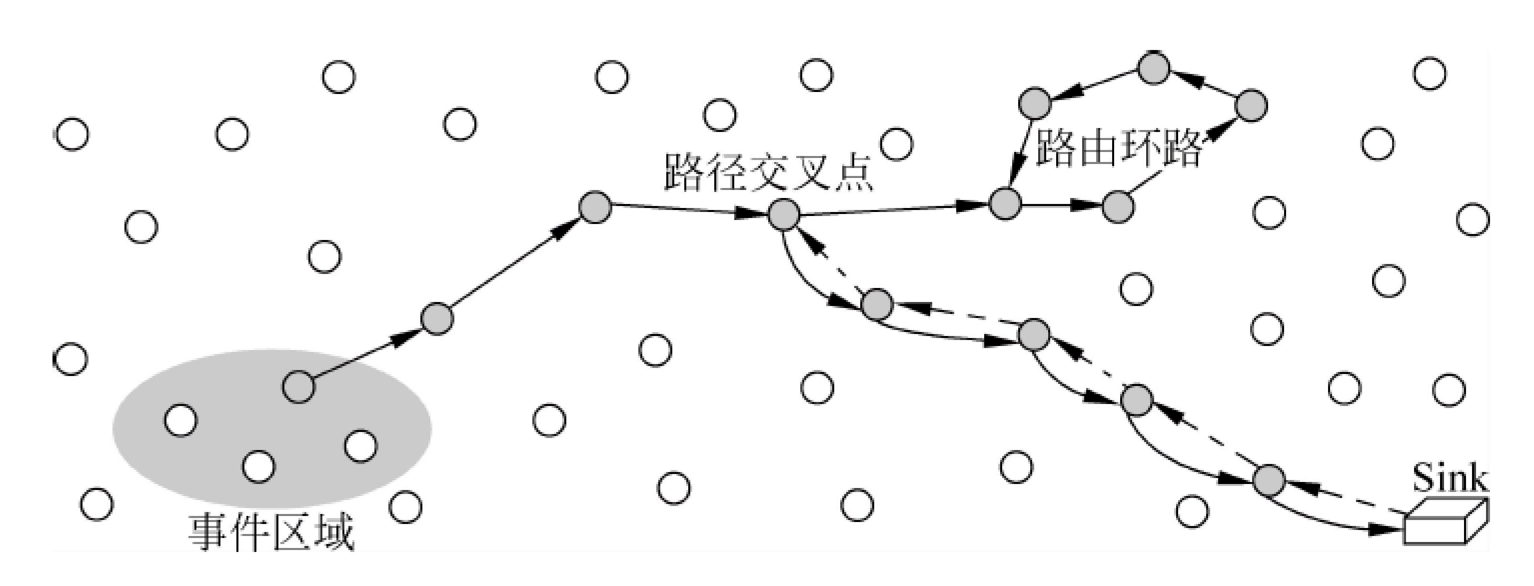


图 5.10　Rumor 路由原理

5.3.4　ZigBee

ZigBee 是一种 Ad-hoc 近距离无线通信技术，由 2002 年 11 月成立的 ZigBee 联盟负责开发，底层协议采用 IEEE 802.15.4 标准，2005 年 3 月发布了 ZigBee 1.0 规范，传输速率最高为 250Kb/s。运用 ZigBee 构造的 WSN 是物联网系统实现的重要技术手段之一。

ZigBee 名称来源于蜜蜂所跳的之字舞（zig-zag），蜜蜂通过这种神秘的"舞蹈"，向同伴

传递花粉所在方位的信息,仿佛构建起了蜂群的通信网络。另外,有人将 ZigBee 字头谐音译为紫蜂,与蓝牙(BlueTooth)遥相呼应,因为两者都是近距离无线通信技术的优秀代表,虽略显做作,但不乏趣味。

ZigBee 技术的特点是近距离、低复杂度、自组织、低功耗、低速率、低成本,主要适用于自动控制和远程控制领域,通信模块便于嵌入各种设备。

为了实现超低功率损耗,ZigBee 采取了一些有效的技术手段:减小报头(地址和其他控制信息)的长度;缩短收发任务周期;采用休眠和断电等功率管理机制;限制在 30m 通信范围之内。这样,ZigBee 网络所需功率一般仅相当于蓝牙组网的 1%,使电池寿命大大延长。

ZigBee 定义了两种设备:完全功能设备,实现全部协议栈,能够与其他各种类型结点互连;简化功能设备,实现简化协议集,在简单的拓扑结构中(星型或点到点)只能作为端结点。另有一个特殊的网络协调器,承担新设备的关联、信标的传输等网络管理任务。

ZigBee 规范包括高层协议栈和底层协议栈(如图 5.11 所示)。

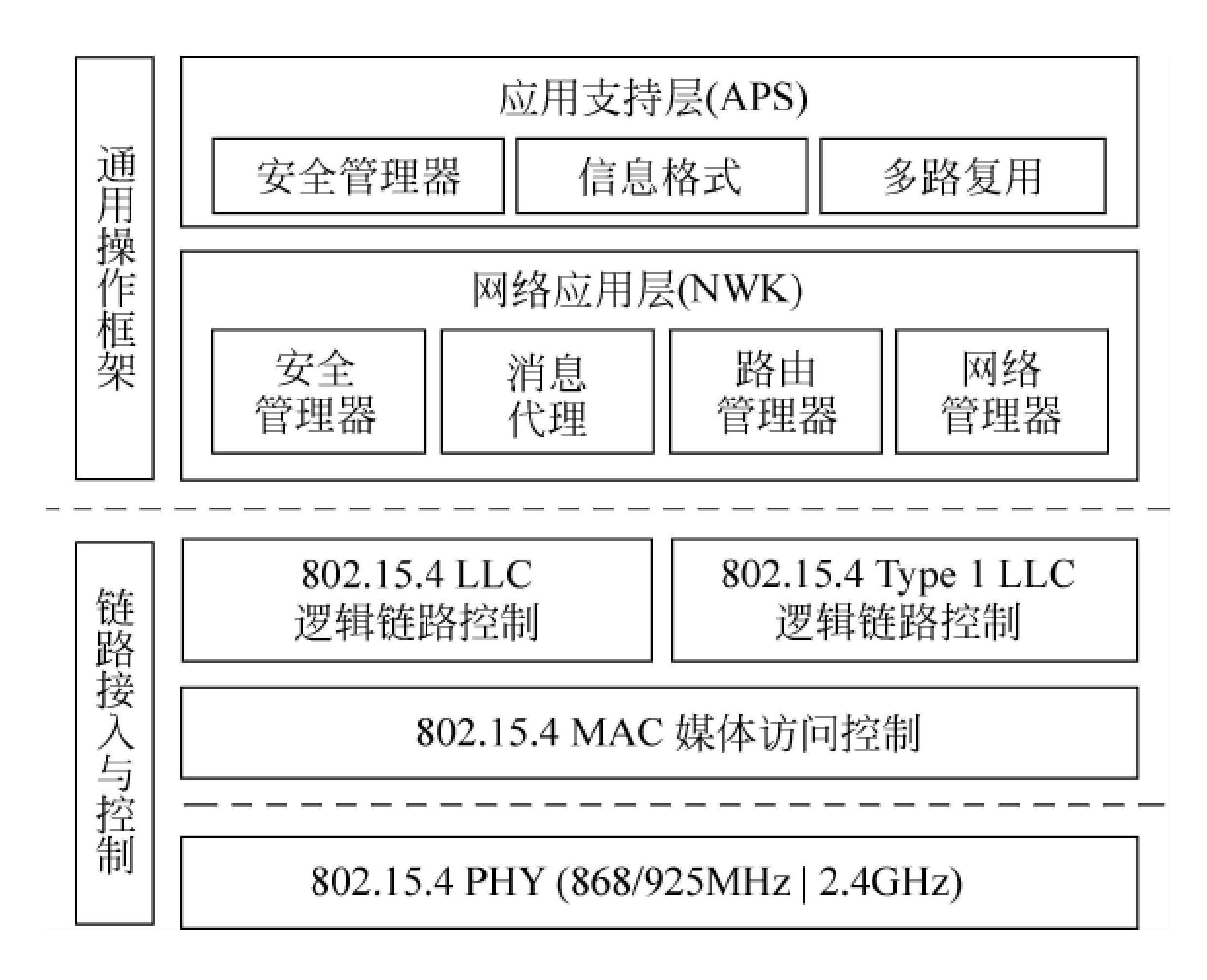


图 5.11　ZigBee 协议栈

高层协议栈由应用支持层和网络应用层所组成,是被称为通用操作框架(General Operating Framework,GOF)的综合层,维护设备描述、网络地址、事件记录、数据格式和其他信息。应用层的应用模型可根据需要定义为支持特定的模式,例如,照明应用模型包含光线亮度等级表示、光覆盖范围的传感器、负载控制器开关和调节器等。网络层主要负责网络的启动、关联、设备地址分配、网络安全、路由等,可支持多层次网络拓扑结构,最多可达 64 000 个结点。

底层协议由物理层、MAC/LLC 所组成,遵循 IEEE 802.15.4 协议,针对实时性要求高的应用进行了优化,使设备唤醒速度快、网络连接时间短。

ZigBee 网络的设备接入物理信道由 TDMA 和 CSMA/CA 相结合进行控制。如图 5.12 所示,ZigBee 采用了超帧(super-frame)结构,一个超帧包括作为起始和结束的信标(beacon frame)和 16 个时隙。

信标由 ZigBee 网络中的协调器在预先定义的 15～252ms 时间间隔中传输,用以实现设备间的识别和同步。信标消息碰撞的可能性较小,不易受 CSMA/CA 的影响。

16 个时隙被划分为两个接入阶段：一个是基于竞争的接入阶段，设备使用 CSMA/CA 算法来确定能够传输数据的时机；另一个是无竞争接入阶段，设备使用由网络协调器分配的 TDMA 保护时隙。预定的信标时间间隔和保护传输时隙的结合，可允许 WSN 结点在休眠期内节约能量，仅当有信标登记或使用保护传输时隙时激活。

结点地址可以采用 64b 全地址，也可采用 16b 短地址格式。

尽管 ZigBee 与 WiFi、BlueTooth 等都使用 2.4GHz 的 ISM 频段，但由于一般的 ZigBee 设备的任务周期很短，所以具有较好的抗干扰性。即使潜在的干扰产生影响，退避机制、ACK 确认的重传机制等技术的运用，将使 ZigBee 设备可以从发送失败中恢复。另外，低任务周期、低数据容量意味着 ZigBee 设备不会产生严重的叠加干扰，不容易对 WiFi、BlueTooth 等网络产生影响。

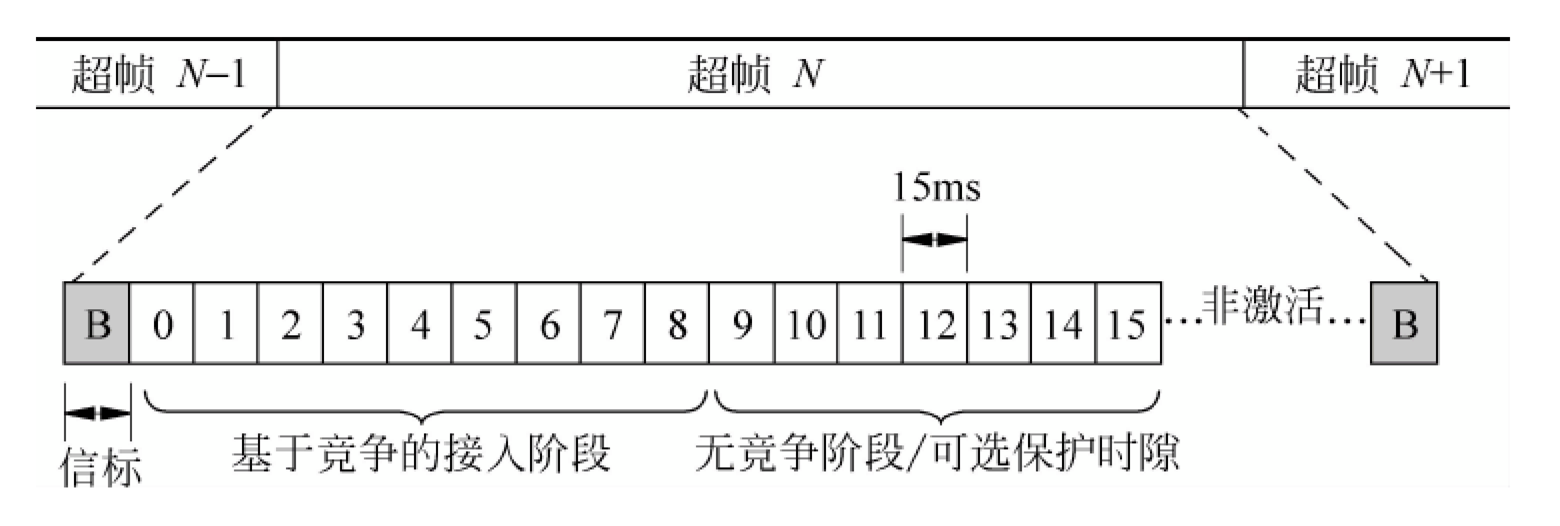


图 5.12　ZigBee 超帧结构示意

CHAPTER 6

第 6 章 宽带网络协议

6.1 宽带网络概述

宽带网络（Broadband Network）是相对于窄带网络（Narrowband Network）而言，差别主要在于通信带宽的高低，但无法一概而论。过去以 T1/E1 速率为所谓的分界线，如今小于 10Mb/s 已经很难称为宽带。

根据 ISDN 定义：B 信道速率为 64Kb/s，用于传输数据；D 信道速率为 16Kb/s 或 64Kb/s，主要用于传输信令。基本速率接口（Basic Rate Interface，BRI）为 2B＋D＝144Kb/s。T1/E1 是主速率接口（Primary Rate Interface，PRI），E1（欧洲标准）为 30B＋D＋64K＝2.048Mb/s，T1（美国标准）为 23B＋D＋8K＝1.544Mb/s。

宽带网络通常指 WAN 和 MAN，因为远程高速通信具有较大难度，需要采用特定的网络协议和组网技术。为了避免流量瓶颈效应、有利于网络互连和扩展，宽带城域网具有层次化构造的特点。

（1）骨干网（backbone）为核心网络，具有最大的带宽、最好的性能。

（2）汇聚网（convergency network）用于聚合下层网络的流量，并通过骨干网进行转发，同时使接入点的部署更加广泛、合理。

（3）接入网（access network）是最下层（外层）的网络，带宽较小，以向用户提供接入服务为目的，如 ISP 网络。

虽然宽带网络未必能与 Internet 划等号，然而，宽带网络主要是为 Internet 或下一代 Internet（Internet 2/NGI）服务，用于 Internet 主干网络、子网互连以及网络接入（即上网）。宽带网络能够提供高速数据传输服务，使网络吞吐能力得到很大提高，为需要占用较大带宽的多媒体网络业务的开展奠定了坚实基础，促进了 Internet 上各类网络应用的繁荣。

如表 6.1 所示，宽带网络类型众多，以分组交换技术为主。绝大部分分组网络由物理层和数据链路层组成，也有少量网络拥有网络层协议（如 X.25 分组交换网）。分组网络组网的主要目的是建立低层通信网，为 IP 等网络层

协议提供虚拟传输链路。数据链路层通过虚电路连接，提供流量控制、差错检测、故障恢复等功能。由于数据链路层的作用，物理信道的特性被很好地掩蔽起来，使上层协议可以不必关心信号传输细节，放心使用分组数据的可靠传输。

所谓数据的可靠(无差错)传输有两个层面的含义：基本的可靠性是指接收方收到的数据一定是正确的，数据的任何差错都会被检测出来，但错误数据可能被丢弃，所以接收方不一定收到所有数据；完全的可靠性是指所有数据都被准确无误地接收，出错的数据都被协议机制(如重发)恢复了。不同的协议可能在不同的层面上对传输可靠性进行保障。

表 6.1　宽带网络技术分布简表

<table>
<tr><th rowspan="2">网络
速率/(b/s)</th><th colspan="2">LAN</th><th colspan="4">MAN/WAN</th></tr>
<tr><th>有线</th><th>无线</th><th>有线</th><th>无线</th><th>专线</th><th>拨号</th></tr>
<tr><td>N×64K</td><td rowspan="2">SDLC
HDLC</td><td rowspan="2">BlueTooth</td><td rowspan="2">PSDN/X.25
FR</td><td>CDPD
GPRS/1x</td><td rowspan="3">DDN
xDSL</td><td rowspan="2">PSTN
ISDN
ADSL
Cable-M</td></tr>
<tr><td>T1/E1</td><td>Microwave
3G</td></tr>
<tr><td>10M</td><td>Ethernet
Novell</td><td rowspan="2">WLAN
(WiFi)</td><td rowspan="3">MPLS</td><td rowspan="2">B3G
WiMax
LMDS
MMDS
Satellite</td><td rowspan="3"></td></tr>
<tr><td>100M</td><td>FastEthernet
FDDI
Token Ring
Token Bus</td><td rowspan="2">ATM
Sonet
SDH
Optical
WDM</td></tr>
<tr><td>≥1G</td><td>GigaEthernet</td><td></td><td></td></tr>
</table>

表中部分缩写含义：

FDDI——Fiber Distributed Data Interface，光纤分布式数据接口(网)

CDPD——Cellular Digital Packet Data，蜂窝式数字分组数据网

Sonet——Synchronous Optical Network，同步光纤网络

SDH——Synchronous Digital Hierarchy，同步数字体系(网)

WDM——Wavelength Division Multiplexing，波分复用

PSTN——Public Switch Telephone Network，公共电话网

6.2　快速分组交换协议

6.2.1　FR

帧中继(Frame Relay，FR)属于快速分组交换技术(FPS)，由 ITU-T Q.922 标准定义，又称 LAPF(Link Access Procedures to Frame Mode Bearer Service)，包括数据链路核心协议(DL-CORE)和数据链路控制协议(DL-CONTROL)两个子集。FR 是对 X.25 技术的改良，用于适应高速信道、高速网络的需要。

FR 技术是面向连接的，但淡化了交换设备上的层次概念，将数据链路层和网络层进行了融合。融合的目的一方面减少了协议处理时间和层次之间的接口处理时间，另一方面通过对融合功能的分析，针对高可靠信道简化冗余项。

被FR精简的协议机制包括：不进行帧重发等完全差错控制，只提供有限差错检测，进一步的恢复操作交给上层协议完成；不对数据帧进行确认；不具备流量控制能力；呼叫控制与用户数据信道分离，通过专门的信令完成，相当于FR核心部分不提供连接建立功能，一般可认为是PVC方式。

如图6.1所示，FR帧头就是HDLC协议的地址字段(A)和控制字段(C)的组合，一般占2B，可扩展到4B，用EA字段进行指示，EA=0表示下一个字节仍然为帧头，EA=1表示帧头结束。

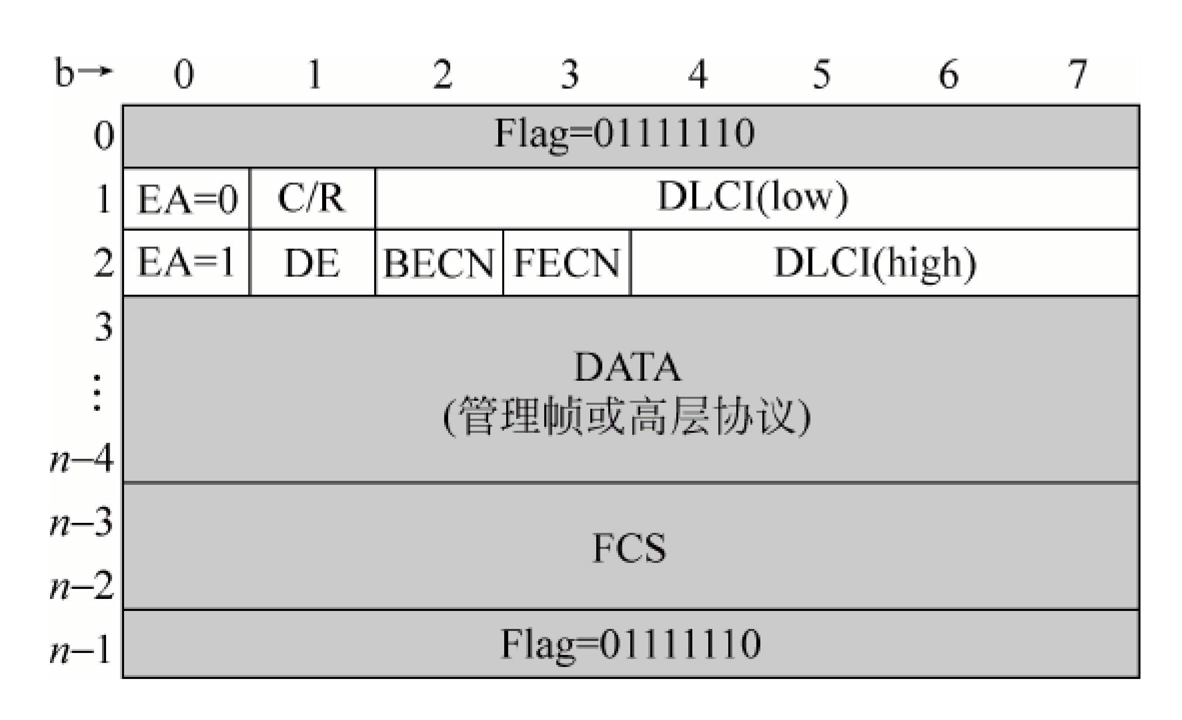


图6.1 FR帧格式

命令/响应(C/R)是由上层实体指明本帧的属性。丢弃指示(DE)置位时表明本帧在网络出现拥塞时可以优先丢弃。信息字段(DATA)可以承载可变长度的用户数据，包括FR管理协议帧、IP等。

最重要的字段为**数据链路连接标识符**(Data Link Connection Identifier，DLCI)，两部分DLCI合起来共占10b，用于标识使用的数据链路。DLCI通常为静态指定和手工配置。FR交换机收到帧头部分的DLCI后，可立即进行交换处理(查询转发表等)，并迅速转入发送，据此可实现高效率的切入式转发。

由于FR不提供帧的重发恢复，也不支持窗口式流量控制机制，只要信道允许，发送方可以连续发送数据，这固然提高了通信效率，但对FR网络造成很大压力，因为网络需要汇集来自各个发送方的数据流，容易引起拥塞。为此，FR设计了一种网络拥塞控制机制，协助交换机和终端对流量进行调整，以避免拥塞后大量数据丢失的情况。FR在帧头中设计了两个二进制位：正向拥塞通知(FECN)和反向拥塞通知(BECN)，均由网络交换设备置位，分别通知接收方与本帧同方向和反方向的传输发生拥塞现象(或趋势)，并可能继续拥塞。

采用这一机制，当FR交换机发现拥塞或接近拥塞时，用FECN通知接收方可能已经发生数据丢失，更重要的是用BECN通知上游的交换机或终端，应减小发送流量，防止拥塞程度的加剧。

FR具有按需分配带宽的特点。用户付费购买承诺信息速率，当突发性数据发生时，在网络允许的范围内，可以使用更高的峰值速率。

FR可配合DDN、ATM网络向用户提供2Mb/s以内的中低速接入服务。尤其ATM技术的光纤信道、接口设备、高速带宽的费用十分昂贵，限制了用户使用，可利用FR与ATM技术的相似性，提供以FR方式接入ATM主干网的服务，这样更为经济和有效。

6.2.2　ATM

异步传输模式(Asynchronous Transfer Mode,ATM)是一种数据链路层协议,目的是提供支持 QoS 的高速数据交换。

如图 6.2 所示,ATM 的协议参考模型分为三个层次。第一层次提供信道驱动和输入输出,相当于 OSI 的物理层;第二层次为核心的 ATM 协议层,与提供上层协议适配的第三层次 AAL 一起,共同构成 OSI 的数据链路层。

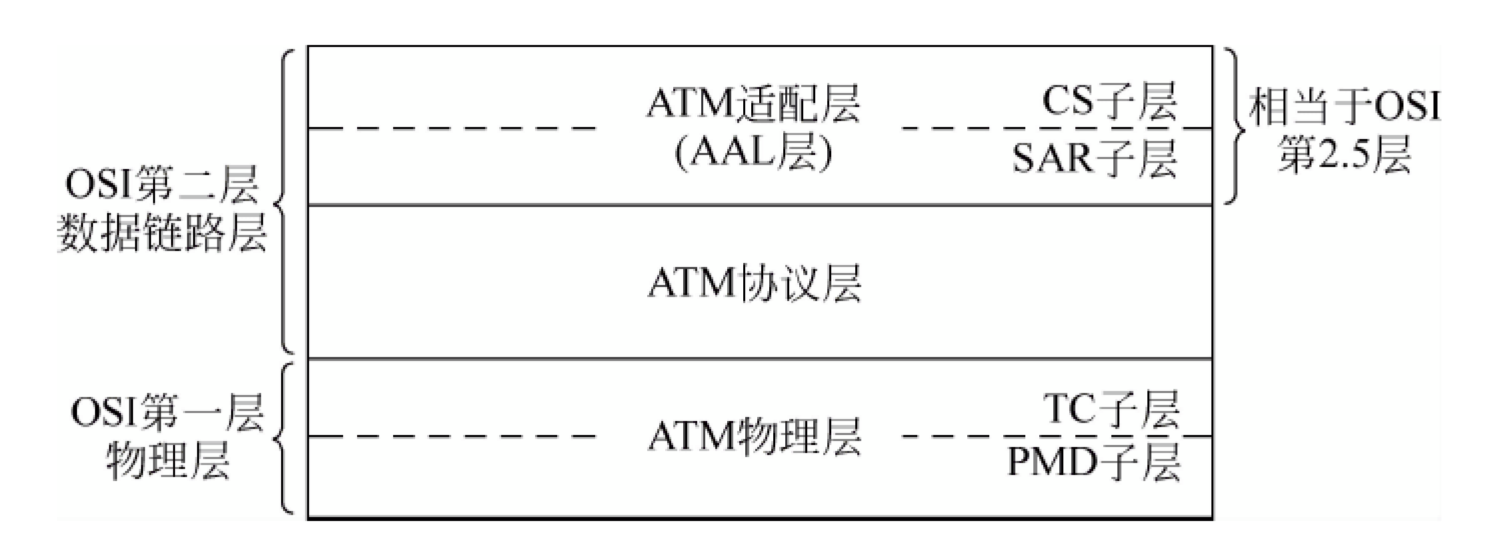


图 6.2　ATM 协议参考模型

ATM 多采用高可靠的光纤传输,支持多种带宽,能适应不同业务需要,可以为 IP 等多种网络层协议提供高质量的数据交换功能,如 IPoA(IP over ATM)。

一项优秀的技术未必能带来广泛的应用。受制于 ATM 设备(包括网卡、交换机、光纤和光传输器件等)相对高昂的成本以及维护管理的复杂性,在与 Ethernet 技术的抗衡中,ATM 节节败退,应用逐步退缩到比较有限的领域。尤其是 Ethernet 进入 1Gb/s、10Gb/s 乃至 1Tb/s 的超高带宽阶段,ATM 又失去了在速率方面的优势,竞争力被进一步削弱了。

ATM 物理层由 PMD 和 ST 两个子层构成。

物理媒体相关(Physical Medium Dependent,PMD)子层负责传送和接收比特流,包括线路传输编码和解码、比特定时、光电转换等。对于双绞线、同轴电缆、单模或多模光纤,ATM 配备相应的 PMD 子层。

传输会聚(Transmission Convergence,TC)子层实现比特流和信元流的转换,包括速率适配(空闲信元插入)、信元定界与同步、传输帧的产生与恢复等。TC 子层的适配使 ATM 协议与传输媒体的特性无关。典型的 TC 子层就是 Sonet/SDH。

SDH 采用同步传输模式,传送的比特流被划分为一个个固定时间长度的 TDM 帧(就是有严格时钟同步的时间片),例如 OC-3 速率标准的 STM-1 帧。但是,ATM 信元可按需插入到 SDH 帧中,每个用户发送的信元在帧内的相对位置并不要求固定,如果一个用户需要发送多个信元,可连续不断地发送,只要 SDH 帧有空位,就可插入这些信元。这属于异步时分复用方式,正是 ATM 名称中“异步”二字的含义。

如图 6.3 所示为 ATM 信元插入 STM-1 帧的方法。STM-1 帧共有 2430B(19 440b),被组成 9 行×270B 的结构。每行的前 9B 为传输开销和指针(指示跨区域边界的信元的位置),其余 9×261B 为有效载荷(净负荷)空间,可以首尾相接依次插入属于不同用户的 ATM 信元。

由于 OC-3/STS-3 速率每秒发送 8000 个 STM-1 帧,即:8000frame/s×2430B/frame×8b/B=155.520Mb/s,简写为 155Mb/s,是 ATM 最常见的速率标准。ATM 另外较常用的

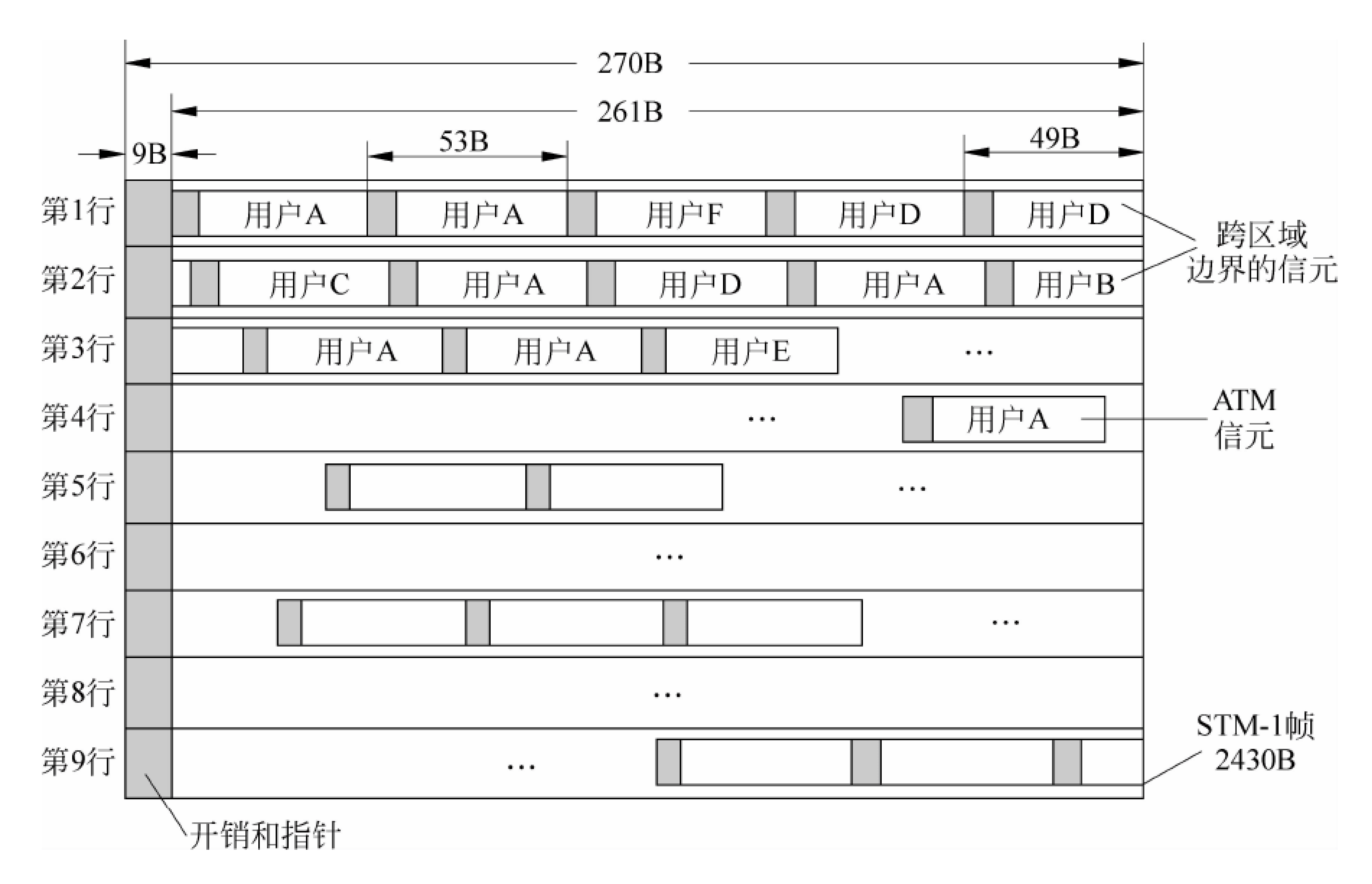


图 6.3 ATM 信元与 STM-1 帧示意

速率标准有 OC-1/STS-1 的 51.840Mb/s、OC-12/STS-12 的 622.080Mb/s。

ATM 协议数据单元是固定 53B 长度的**信元**(cell),如图 6.4 所示,数据字段为 48B,信头(5B)分为两种,一种用于用户与网络接口(User-Network Interface,UNI),另一种用于结点之间,即网络与网络接口(Network-Network Interface,NNI),两种信元在头两个字节上有所不同。

通用流量控制(General Flow Control,GFC)为 4b 字段,用以在共享信道上进行接入流量控制(类似 MAC 层功能),在点对点线路上固定为 0。

虚通道标识(Virtual Path Identifier,VPI)和虚通路标识(Virtual Channel Identifier,VCI)字段用于标识逻辑连接、实现路由选择。如图 6.5 所示,一条传输链路可以由多个虚通道(VP)组成,每个 VP 则可以包含多个虚通路(VC)。UNI 有 8b 的 VPI(256 个),NNI 因为承担更多的通道交换任务的关系,需要更多通道标识空间,所以有 12b 的 VPI(4096 个),两者均有 16b 的 VCI,即每 VP 可分配 65 536 个 VCI。有些 VPI/VCI 组合已定义为指定用途,例如 0/5 为默认信令连接。

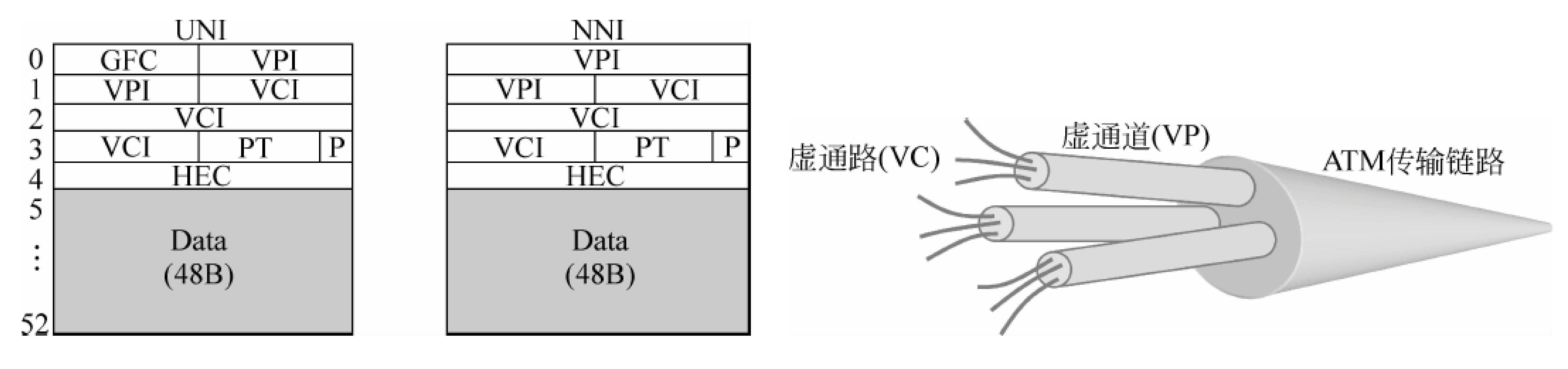


图 6.4 ATM 信元格式　　　　图 6.5 ATM 的 VP 和 VC 示意

载荷类型(Payload Type,PT)为 3b 字段。第一个 $bit_1=0$ 表示载荷是用户数据,否则就是信令数据;第二个 $bit_2=1$ 表示网络拥塞,由 ATM 交换机负责填写,用于流量工程;第

三个 $bit_3=1$ 表示载荷携带的是最后一个分片数据。

信元丢弃优先级(Cell Loss Priority,CLP)为 1b 字段(即信元格式图中的 P 字段),用于 ATM 网络 QoS 管理中的流量控制机制。当网络负担很重即将出现拥塞时,ATM 交换机优先考虑丢弃 CLP=1 的信元,以缓解网络压力。除了端系统可指定 CLP 值,交换机也可将违反通信量合约的信元 CLP 值置 1,称为打标记(tagging)。

首部差错控制(Header Error Control,HEC)为 8b 字段,对首部前 4B 进行 CRC 校验。HEC 是由物理层产生和进行检验操作的。

ATM 通常采用 PVC 静态配置方式,但也可通过 ATM 的信令协议动态维护逻辑连接。ATM 的信令协议由 ATM 论坛的 UNI 3.1 和 UNI 4.0、ITU-T Q.2930、Q.2931 和 Q.2110 定义,有较大的复杂性,实际应用很少。

图 6.6 所示为 ATM 建立连接、传输数据和拆除连接的过程。连接建立和释放使用信令消息(Signaling Message),信元中 CLP=1,可以由一个或多个信元来传送;传输数据采用普通的信元,CLP=0。信令消息的内容包括消息类型、长度、QoS 参数、对该连接指派的 VPI/VCI 等。

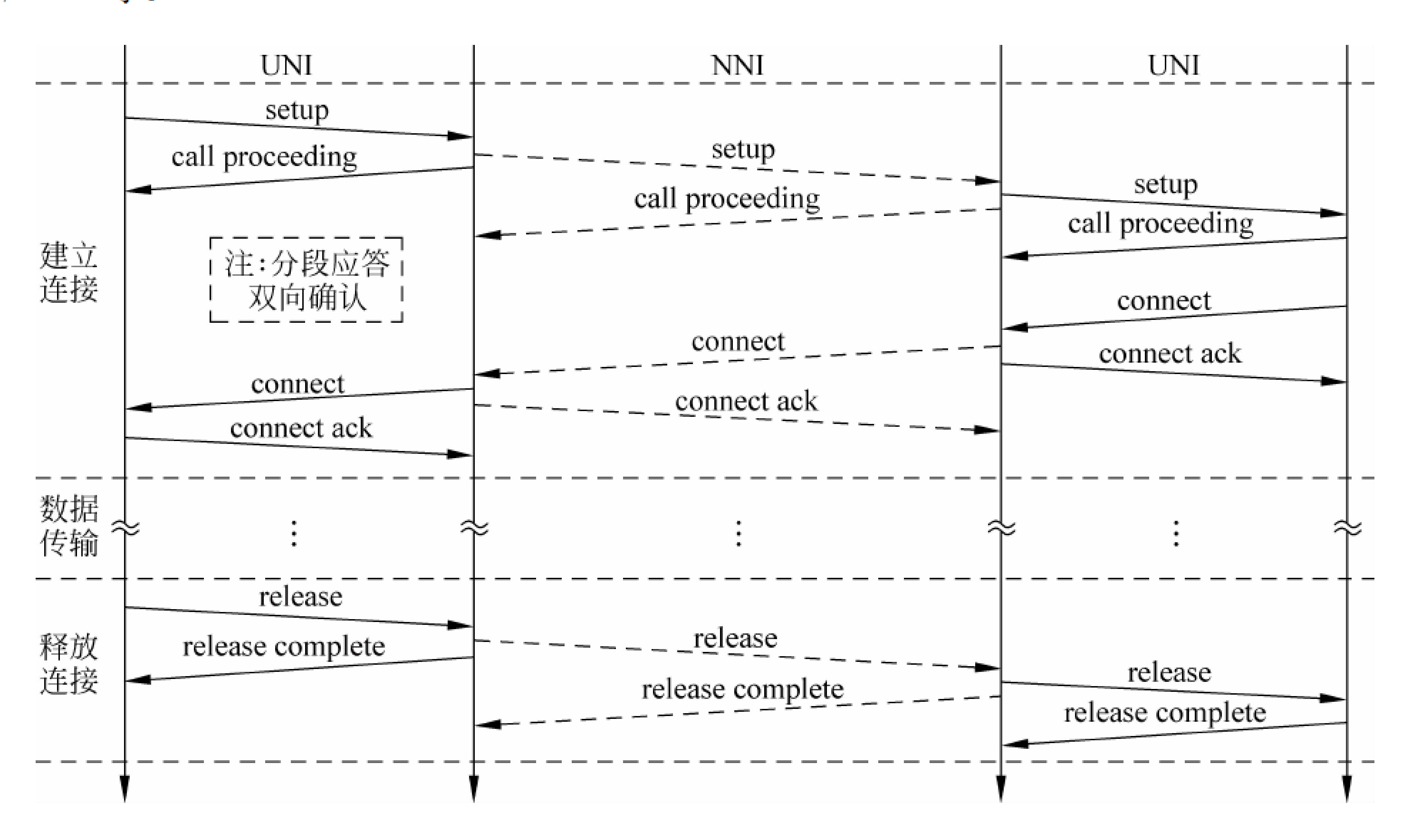


图 6.6　ATM 逻辑连接流程

ATM 适配层(ATM Adaptation Layer,AAL)包含 CS 和 SAR 两个子层,用于增强 ATM 所提供的功能,向上层提供各种服务。

会聚子层(Convergence Sublayer,CS)可面向不同的应用,向上层协议提供服务访问点(SAP),使不同的 AAL 用户(不同协议、不同进程)可调用 ATM 的数据传输功能。CS 的协议数据单元为 CS-PDU。

拆装子层(Segmentation And Reassembly,SAR)负责将 CS-PDU 拆分成 48B 单元,交付给 ATM 协议进行传输;在接收时,SAR 则负责将 48B 有效载荷装配成 CS-PDU。拆分或拼装的最后一个单元的信元 PT 字段中 $bit_3=1$(即 PT=0*1),其他单元的信元 PT 字段中 $bit_3=0$(即 PT=0*0)。

AAL 只存在于 ATM 端系统中。为支撑不同应用,AAL 定义了 5 种类别(class),命名

为AAL1～AAL5,其中简单而高效的AAL5应用最广。AAL5的CS-PDU由用户数据(加上报尾后填充为48的整数倍)和8B的控制报尾组成。报尾包括:2B保留字段;2B长度字段,指明用户数据的有效长度(以B为单位);4B的CRC校验字段。

为了便于通信量管理,ATM服务按照比特率的特点划分为5个种类(Category):衡定比特率(Constant Bit Rate,CBR)、实时可变比特率(real-time Variable BR,rt-VBR)、非实时可变比特率(non-real-time VBR,nrt-VBR)、不指明比特率(Unspecified BR,UBR)和可用比特率(Available BR,ABR)。

ATM网络由ATM交换机(ATM switch)构成。ATM交换机的主要功能是实现ATM协议(包括信令协议),并完成高效率的信元交换。实际上后者才是最核心、最不可或缺的职能,因为很多ATM交换机仅支持PVC静态设置。为此,ATM交换机还需要支持管理功能,用于配置、诊断和维护。

ATM交换机的关键部件是交换机构(switching fabric),交换机构维护一张端口和VPI/VCI转换表。如图6.7所示,交换机构根据逻辑连接确定端口之间的连接关系,来自一个端口的信元,根据VPI/VCI检索转换表,可以确定连接到另一个转发端口,同时替换成该端口下新的VPI/VCI值。

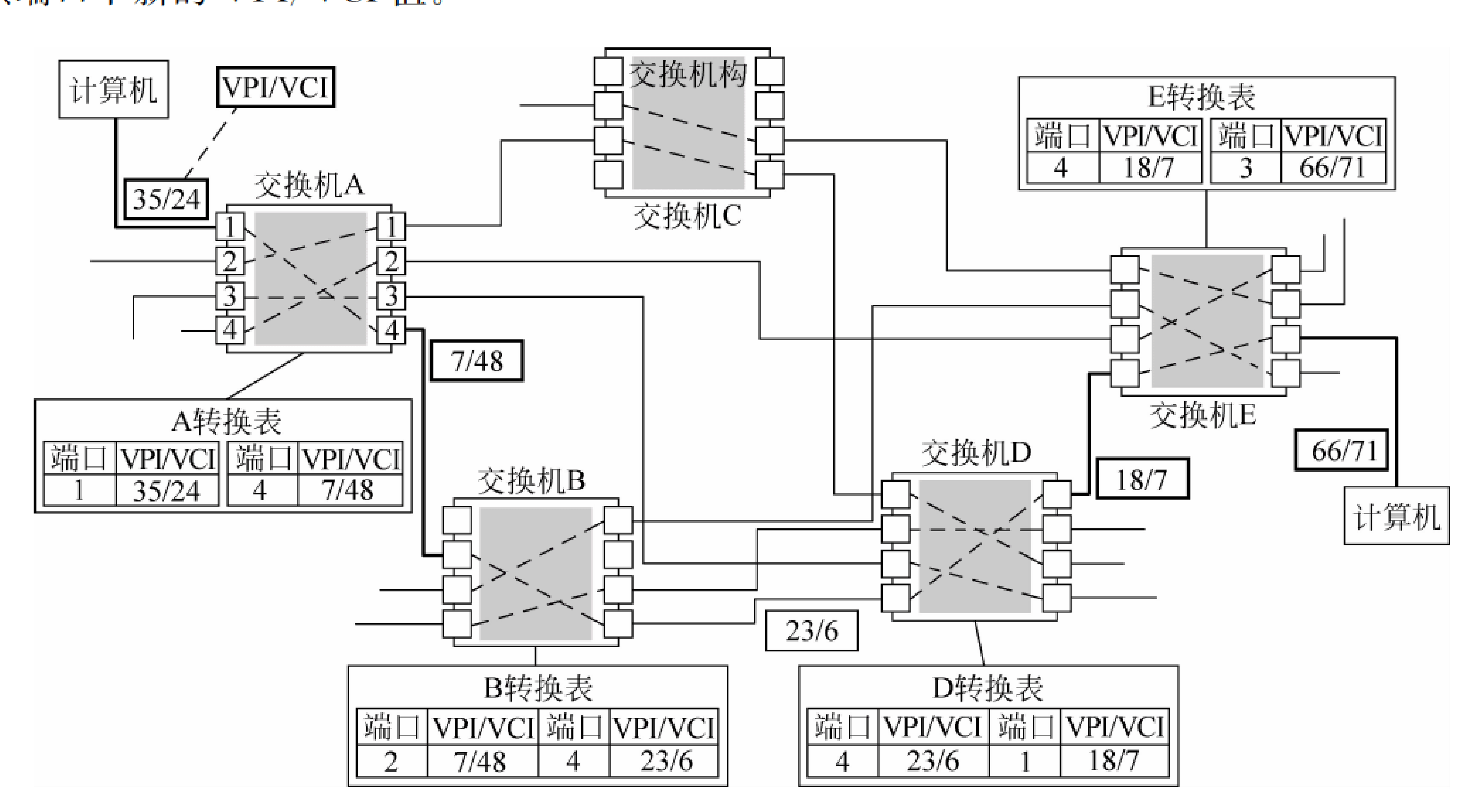


图6.7 ATM交换原理

仅对VPI进行识别和交换的特殊ATM交换机被称为交叉连接交换机(cross-connect switch)。

6.2.3 MPLS

多协议标记交换(Multi-protocol Label Switching,MPLS)是一种在开放的通信网上提供数据高速、高效传输的协议技术(RFC 3031/3032)。

受到ATM技术的启发,MPLS着眼于在无连接的网络中引入连接模式,改善网络层路由技术的性能,提高网络的可伸缩性,并在路由服务方面提供更大的灵活性,在不改变网络转发方案的情况下,允许添加新的路由,从而减少网络的复杂性,能够兼容现有各种主流网

络技术。

在早期的邮政体系中，信件和包裹的传递是依靠查看“收件人地址”来进行的，从一个省（国家）发往另一个省（国家）、从一个城市传递到另一个城市，直到递送给收件人。书写收件人地址使用自然语言，例如中文、英文、日文、韩文……邮局的工作量很大。邮政编码（zip code）的出现极大地提高了邮政系统的工作效率，因为只需识别固定位置上的全国唯一的 6 位数字（例如 200433 就是复旦大学所属区域的邮政编码），就能轻而易举地把信件准确地转交给目的地邮局。邮政编码就好比给信件贴上了一张标准化的、便于识别和操作的数字标签，使信件的中间传递过程与地址书写所用的语言无关，可实现自动化分拣。

MPLS 协议的技术关键之一是引入了标记（label）的概念。标记是一种简短的、易于处理的、不包含拓扑信息、只具有局部意义的信息内容，通常可以用索引直接引用。标记只具有局部意义是为了便于分配。

从 MPLS 标记的含义，可以联想到 ATM 技术中的 VPI/VCI。其实，VPI/VCI 就是一种标记，因此，ATM 实际上也可称为一种标记交换协议。与之类似的标记还有 FR 的 DLCI、X. 25 的 LCN 及 TDM 的时隙等。

MPLS 体系结构描述了实现标记交换的机制，这种技术兼有第二层交换的分组转发技术和第三层路由技术的优点。与第二层（FR、ATM 等）相似，MPLS 给报文分配标记，以便报文能够在基于分组或信元的网络中传输。数据单元携带的长度固定的短小标记告诉传输路径上的交换结点如何处理和转发数据。

MPLS 的这种特性相当于对各种报文进行统一封装（如图 6. 8 所示），并且据此进行数据交换，在满足 QoS 要求的同时，屏蔽了不同协议的不同技术特性；更重要的是，使原来不支持 QoS 的协议也具备了 QoS 支持能力和快速分组交换能力。

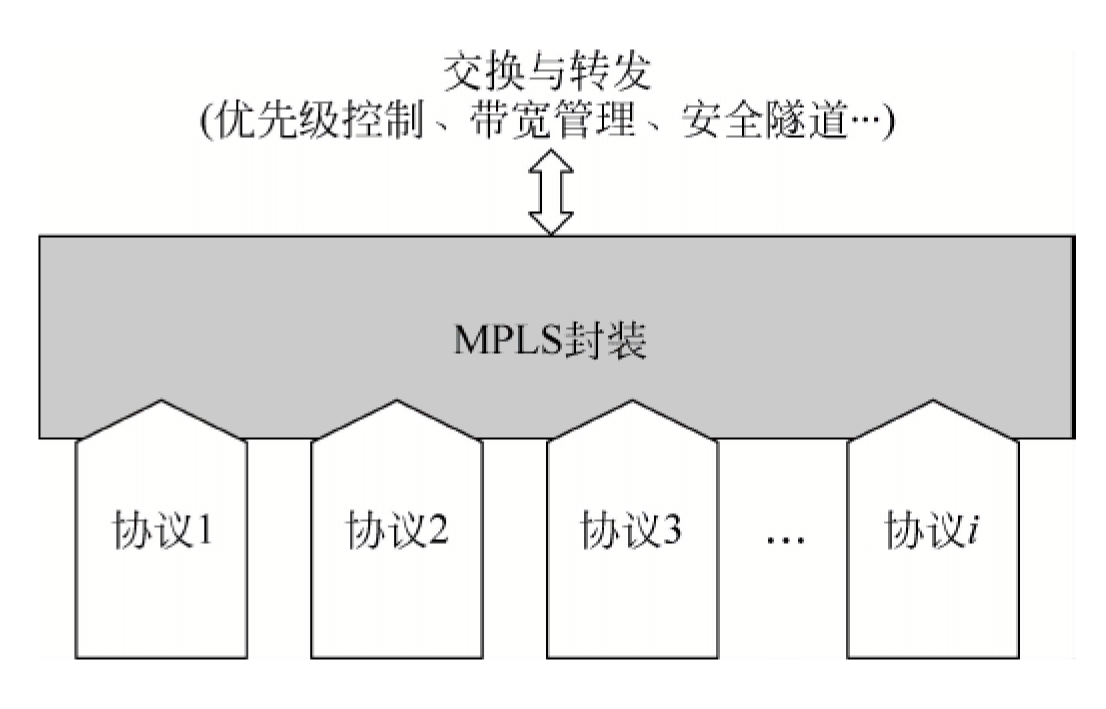


图 6. 8　MPLS 封装多协议示意

MPLS 体系结构被分为两个独立的组件：转发组件（数据层面）和控制组件（控制层面）。转发组件使用 MPLS 标记交换机维护的标记转发数据库，根据分组携带的标记执行数据的转发任务；控制组件则负责在一组互连的 MPLS 标记交换机之间创建和维护标记转发信息。MPLS 技术最初是用来提高网络设备的转发速度而提出的一个协议，由于其在提高网络服务质量方面的良好表现，也成为 QoS 技术的重要标准。

从另一个角度来看，MPLS 是一种面向连接的协议，只是不提供数据可靠传输功能。MPLS 的标记及其实现的功能也是 PVC 的表现形式之一。MPLS 提供的逻辑连接可以是端到端，也可以是点到点，但最终形成一条端到端的连接。

如图6.9所示，MPLS网络由两类路由式交换机构成。

图6.9　MPLS网络组成示意

在网络边缘的结点称为标记边缘路由器(Label Edge Router，LER)，而网络的核心结点为标记交换路由器(Label Switching Router，LSR)或标记交换机。LER结点在MPLS网络中完成IP报文的进入和退出过程，LSR结点在网络中提供高速交换功能。

在MPLS结点之间的路径叫做标记交换路径(Label Switching Path，LSP)。一条LSP可以看做一条贯穿网络的报文隧道。

在LER中，MPLS使用转发等价类(Forwarding Equivalence Class，FEC)将输入的数据流映射到一条LSP上。换句话说，FEC就是定义了一组沿着同一条路径、有相同处理过程的数据包。这就意味着所有FEC相同的报文都可以作为一个流束(stream)映射到同一个标记中。

FEC可以是目的地址与IP地址前缀匹配的报文(相当于普通的路由器)，可以是所有源地址和目的地址都相同的报文，也可以是具有指定的QoS需求的报文。FEC可用于网络流量的负载均衡，为不同的FEC选择不同的转发路径，相比普通的路由协议只能选择单一路由(而且一旦路由收敛路径就固定下来)，MPLS显得更灵活、流量管理更细致。

LER和LSR以及LSR之间通过标记分配协议(Label Distribution Protocol，LDP)通信，LDP负责与路由协议(如OSPF、IS-IS、EIGRP、BGP等)协同工作，为要建立LSP的边界和核心设备分配标记。除LDP外，IETF还支持限制路由的标记分配协议(Constraint Route Label Distribution Protocol，CR-LDP)和扩展资源预留协议(RSVP Extension)。

MPLS网络的数据交换过程如图6.10所示。

首先，进入的MPLS数据报文经过处理后，被赋予标记。大量的分析、定级、过滤等密集型的处理都在入口边界LER上完成。这一步称为MPLS的网络边界行为。

其次，在MPLS网络的内部，MPLS的LSR解读标记，检索标记信息库，用相应的出口(egress)标记对换(swapping)入口(ingress)标记，转发数据报文，因此所有的处理操作仅是根据携带的标记来转发数据报文而已。这一过程称为MPLS的网络核心行为。

最后，在出口边界LER上，标记被从数据报文上剥离，数据报文被转发到最终的目的地。

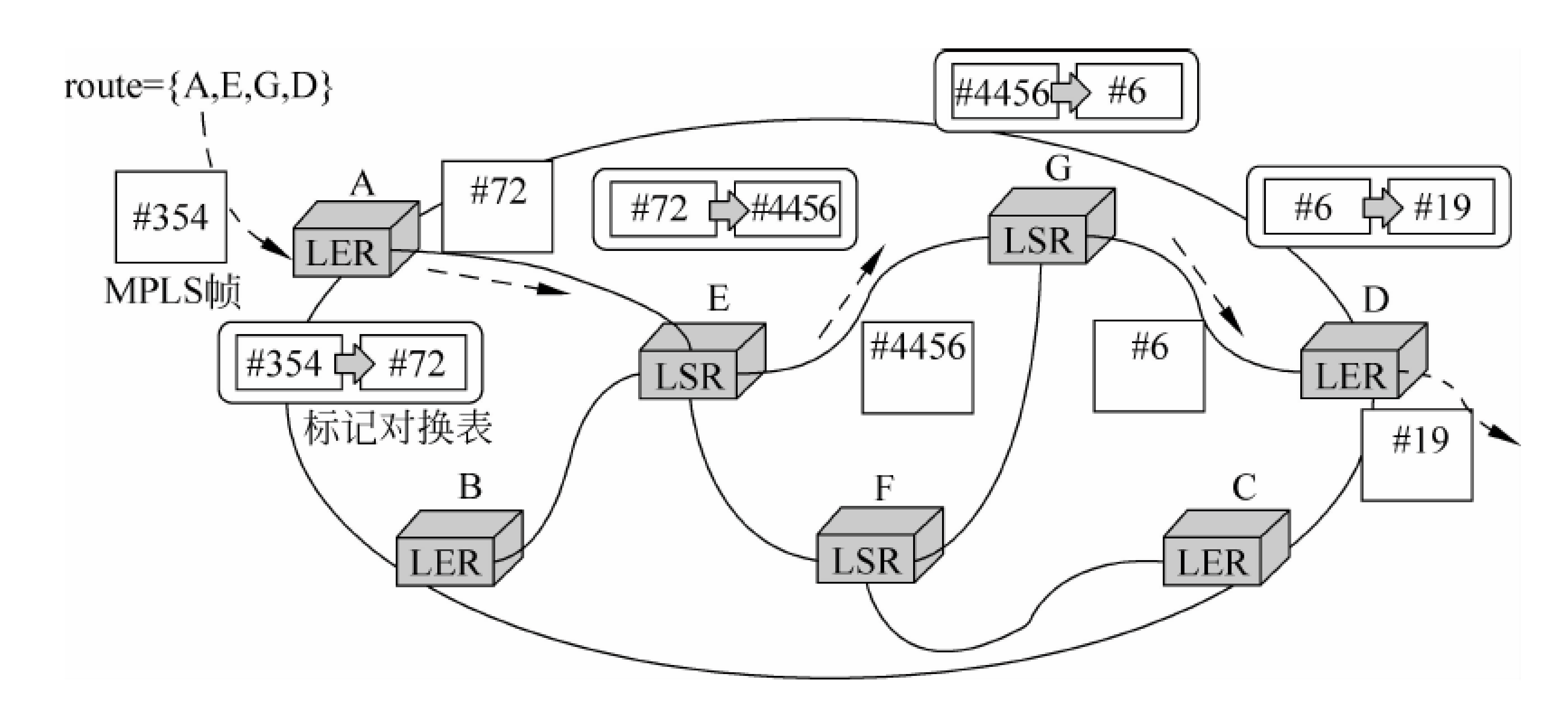


图 6.10 MPLS 标记对换和转发示意

因为所有的标记处理都在边界 LER 上进行，内部 LSR 仅仅做标记交换，数据报文不必在每一跳上都进行处理，这是相比其他传统协议的主要区别和优势，也是加大网络设备数据吞吐量的关键点。

建立 LSP 的方式主要有两种。

(1) 逐跳路由(Hop-by-Hop Route)。一个 Hop-by-Hop 的 LSP 是所有从源站点到一个特定目的站点的 IP 报文路径树的一部分。对于这些 LSP，MPLS 模仿 IP 转发数据报文的面向目的地的方式，建立了一组路径树。

从传统的 IP 路由来看，每一台沿途的路由器都要检查报文的目的地址，并且选择一条合适的路径将数据包发送出去。而 MPLS 则不然，数据包虽然也沿着 IP 路由所选择的同一条路径进行传送，但是 IP 数据报头在整条 LSP 路径上从始至终都没有被检查。

在每一个结点上，MPLS 生成的树是逐级为下一跳分配标记，而且是通过与对等层交换标记而生成的。交换标记是通过 LDP 的请求以及对应的消息完成。

(2) 显式路由(Explicit Route)。MPLS 最主要的一个优点就是可以利用流量设计"引导"数据包，例如避免拥塞或者满足业务的 QoS 等。MPLS 允许网络的运行人员在源结点就确定一条显式路由的 LSP(ER-LSP)，以规定数据报文将选择的路径。

不像 Hop-by-Hop 的 LSP，ER-LSP 不会形成 IP 树。取而代之，ER-LSP 从源端到目的端建立一条直接的端到端的路径，如图 6.10 所示。MPLS 将显式路由嵌入到标签分发信令协议(如 CR-LDP)的信息中，从而建立这条路径。

MPLS 协议报头格式如图 6.11 所示。

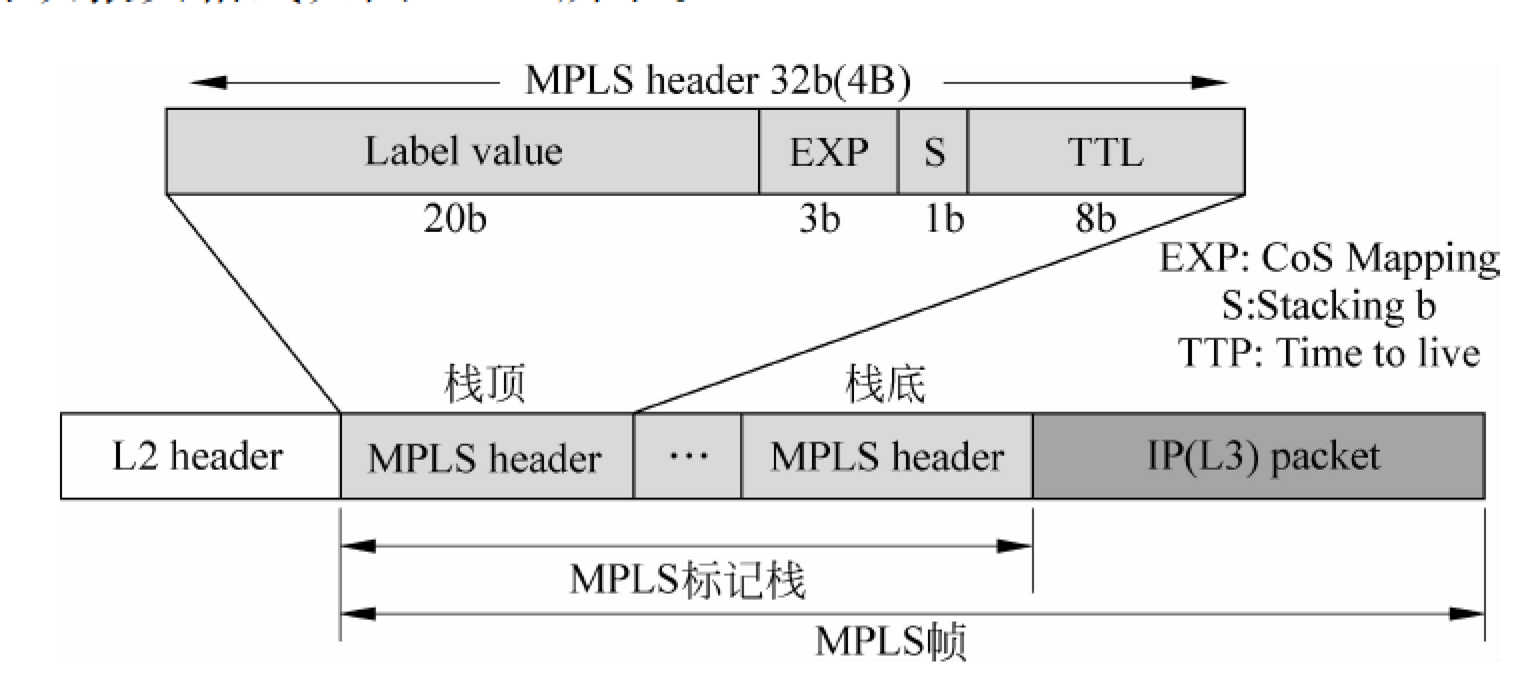


图 6.11 MPLS 报头格式

MPLS 报头中最核心部分为 20b 的标记(label)字段(有时也称为标签),类似 FR 的 PVC 模式下的 DLCI、ATM 协议的 VPN/VCN。

EXP 为 3b 的实验(experimental)字段,可用于 CoS 设定,使转发行为管理的细粒度更高。

TTL(time-to-live)为 1B 的生存时间字段(类似 IP 报文),每经过一个结点 TTL 被减 1,用以防止报文被错误地无限制转发。

S 为 1 比特的堆栈标识,当 MPLS 出现嵌套时(如图 6.12 所示),新的 MPLS 报头被压到堆栈的栈顶,后进先出,其中栈底(或无嵌套)报头的 S 位置 0,否则就置 1 表示还有更多的嵌套报头。

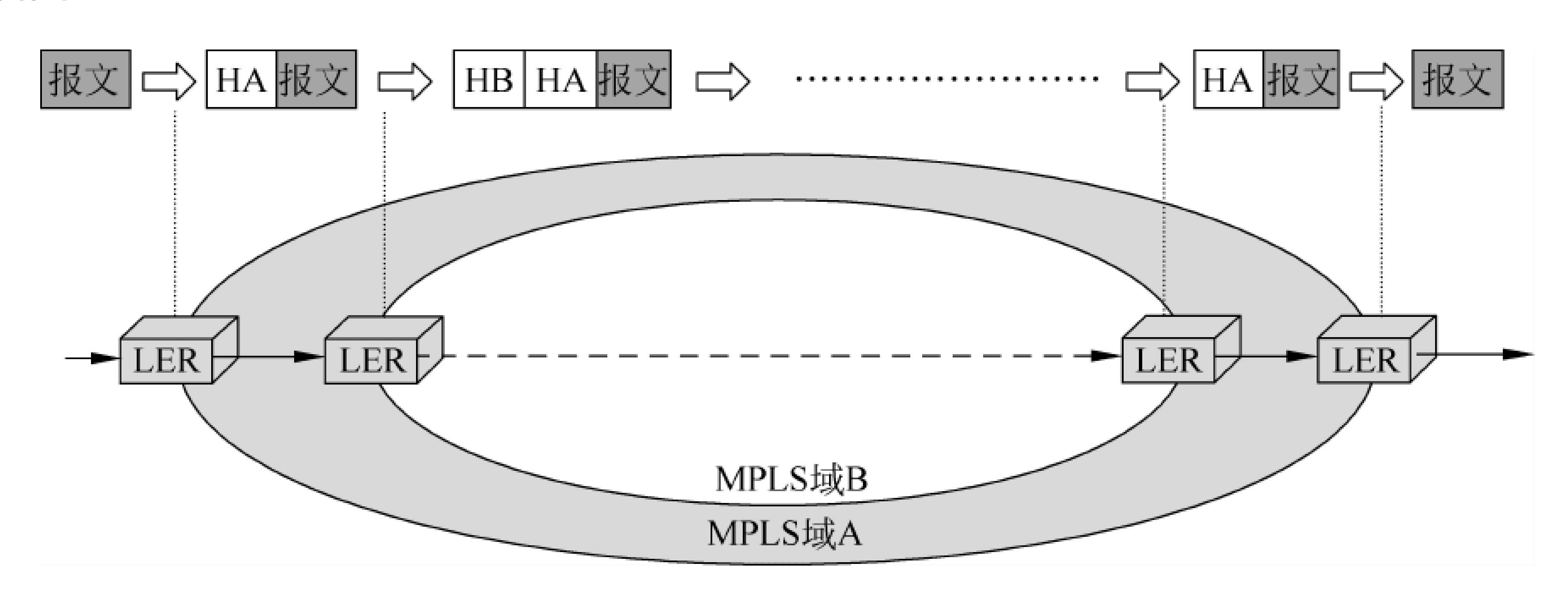


图 6.12 MPLS 嵌套

可见,MPLS 协议非常简单,却十分高效,可以封装不同协议数据单元,达到标记交换的目的。

再以物流行业为例,MPLS 好比快递员,将客户需要传递的东西打包封装在快递袋内,不论是文件、玩具、电器还是化妆品,都以一致的服务方式送货上门,还能根据需要进行急件急送。

MPLS 标记的分配与分发由 LDP 等信令协议完成。LDP 与传统路由算法相结合,进行分配标记、发布 FEC 和标记绑定信息、建立和维护标记对换表和标记交换路径等。

MPLS 标记分发方式有两种:上游按需请求方式、下游按需分配方式。上游按需请求方式是指上游 LSR 为某一个 FEC 向下游 LSR 请求分配标记,下游 LSR 可视具体情况给上游 LSR 分配标记;下游按需分配标记方式则不需要上游请求标记,直接将标记绑定消息后发送给上游。

MPLS 标记分配过程主要包含两种消息:标记请求消息、标记映射消息(即标记分配消息)。图 6.13 所示为标记请求和标记映射(分配)的时序。

MPLS-TE(流量工程)是指为了平衡网络设备的流量,根据数据流量进行显式路由选择的过程,用于提高网络运作效率和可靠性,优化网络资源利用和流量性能。MPLS-TE 通过将流量导入指定的路径来实现其目标。

MPLS-TE 的应用包括以下几方面内容:流量统计分析、流量优化、网络保护和提供 QoS 保障。特别是在流量优化方面,MPLS-TE 显得主动性很强。通过流量分析获得网络流量的分布及其状况,在不同的结点间建立不同的显式路径 LSP,引导流量按照规划的路径进行传播,将原本负担较重、较为拥堵的链路的流量分担到相对空闲的设备和链路上。

在网络保护方面，MPLS-TE 提供一种高性能的自愈恢复机制。Sonet/SDH 恢复时间在 50ms 量级，IP 路由恢复时间达到几十秒，MPLS 恢复时间可以接近 Sonet/SDH 水平，因为 MPLS-TE 采用了类似的链路保护机制，称为快速重新路由（Fast Re-route），为每个链路和结点提供迂回路由，一旦结点或链路发生故障，立即由上游结点检测到并切换到备用路径上。实际应用中可以对网络的关键部分进行保护，以减少资源的冗余度和计算的复杂性。

MPLS 还可以提供有效的 VPN 服务。由于 MPLS 的转发是基于标记的，并不依赖 IP 地址，可解决 VPN 内专用地址复用问题，同时 MPLS 链路的保密性更强。如果 MPLS 选择参与 IP 路由，能够进一步提高部署的灵活性。

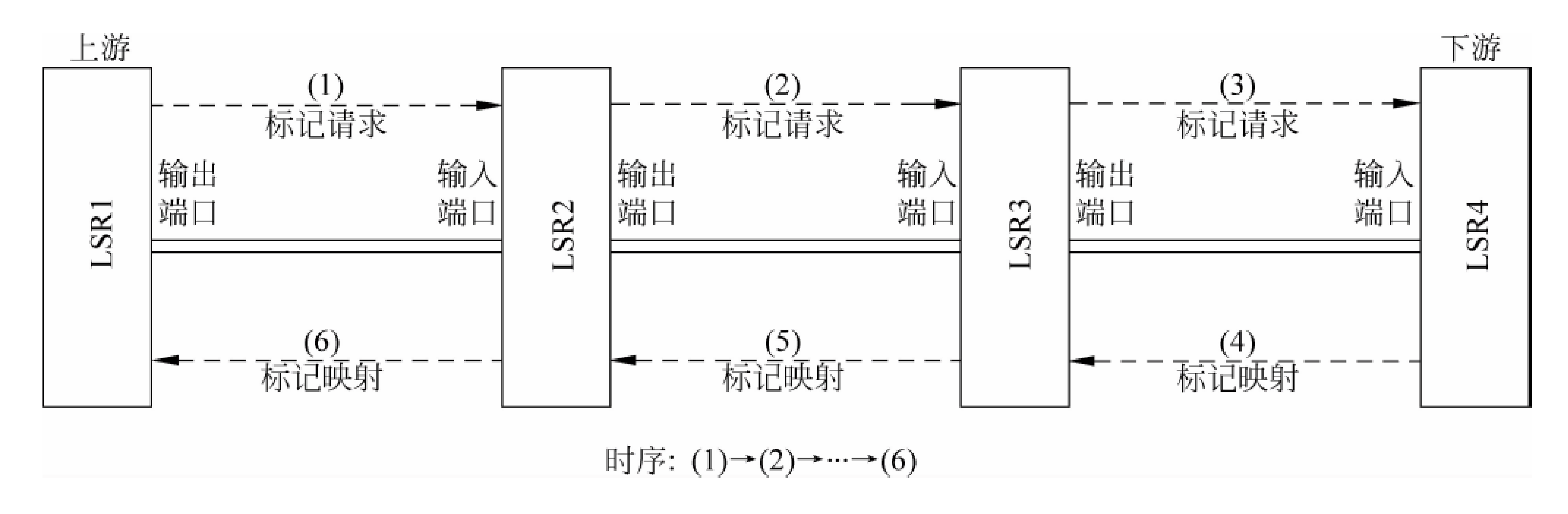


图 6.13　MPLS 标记按需请求过程

6.3　多媒体应用协议

6.3.1　NTP

多媒体应用系统中经常需要保持各通信结点的同步。为了避免采用较为昂贵的卫星定位系统对时机制或更为复杂的时间同步网络，可使用专门的协议，使每个结点的时钟与世界协调时间（Universal Time Coordinated，UTC）同步。

网络时间协议（Network Time Protocol，NTP）是网络上用于计算机对时的常用技术，最新为 V4 版本（RFC 2030），用 UDP 传输报文（端口号 123）。

NTP 协议具有 C/S、对称（symmetric）和广播/组播（broadcast/multicast）三种工作方式，对时精确度在局域网环境下可达 10μs～10ms，在 Internet 上为 100～1000ms。

由于网络传输存在时延，因此 NTP 需要通过测定时间偏移量来调整准确时间。计算机 C 与时间服务器 S 之间的对时协议如下（如图 6.14 所示）。

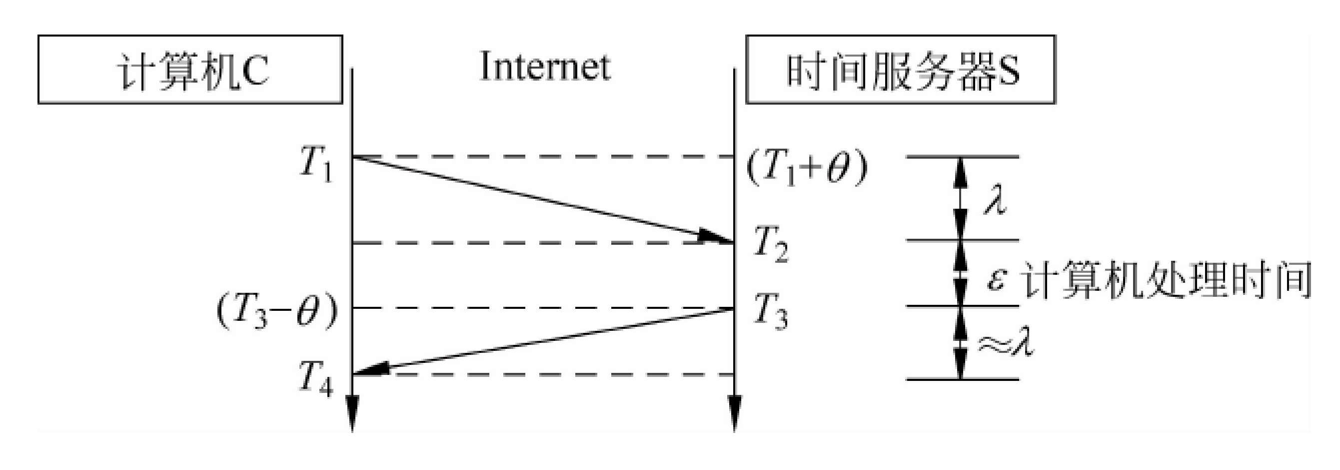


图 6.14　NTP 消息交换过程

(1) C 发送 NTP 消息给 S,消息带有 C 的时间戳 T_1。

(2) S 收到 NTP 消息,加上 S 的时间戳 T_2。

(3) S 发送 NTP 消息,再加上离开时 C 的时间戳 T_3。

(4) 当 NTP 消息回到 C 时,C 记录此时本地时间 T_4。

设标准时间与计算机差为 θ,网络传输报文单程时间需 λ,假定来回报文所需时间相同。根据 NTP 携带的时间戳,列出如下关系方程组:

$$\begin{cases} T_2 = (T_1 + \theta) + \lambda \\ T_4 = (T_3 - \theta) + \lambda \end{cases}$$

解得

$$\begin{cases} \lambda = \dfrac{(T_2 - T_1) + (T_4 - T_3)}{2} \\ \theta = \dfrac{(T_2 - T_1) - (T_4 - T_3)}{2} \end{cases}$$

则本地正确时间=现在时间+θ。

可见关键的 θ 与报文传输时间 λ 和时间服务器处理时间 ε 均无关。但是,由于报文每次穿越网络所需时间并不确定,因此,来回报文时间相等的假定是造成时间误差的主要原因。

除 NTP 外,还有简单网络时间协议(SNTP)、精确时间协议(Presicion Time Protocol,PTP)和通用定时接口(Universal Timing Interface,UTI)等网络时钟同步(对时)技术。

6.3.2 RTP

实时传送协议(Real-time Transport Protocol,RTP)为宽带网络实时应用(如流媒体)提供端到端的数据传送,但不提供任何 QoS 保障。RTP 可以视为应用层的一个子层,工作在 UDP 之上,于 1996 年由 RFC 1889/1890 定义,最新版本为 RFC 3550。

RTP 报文只包含 RTP 数据,而控制由配套的 RTCP 提供。RTP 选择 1025~65 535 之间一个未使用的偶数端口号,而同一次会话的 RTCP 就使用下一个奇数的端口号。5004 和 5005 分别为 RTP 和 RTCP 的默认端口号。RTP 报文格式如图 6.15 所示。

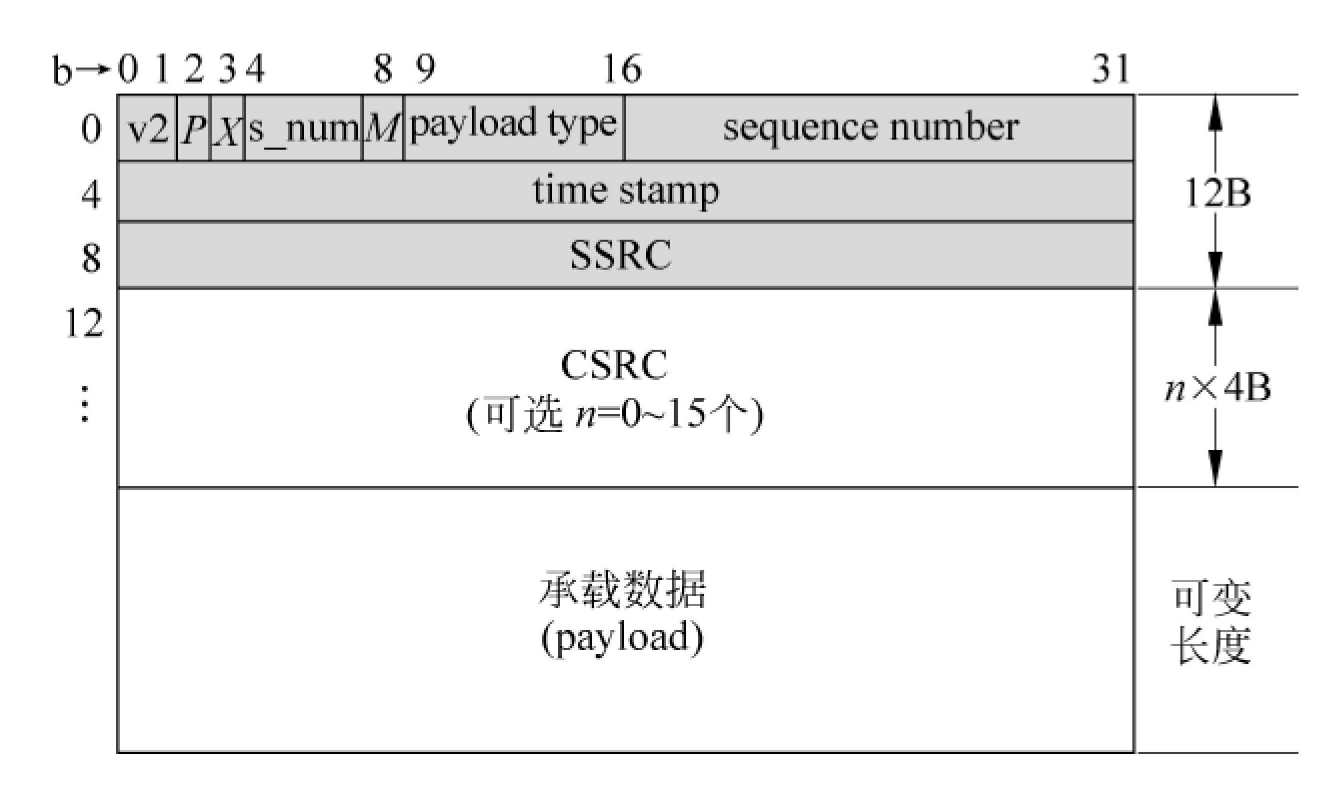


图 6.15 RTP 报文格式

版本号字段为 2b,版本 2 值为 2;填充 P 字段为 1b,若 $P=1$,表示出于加密等需要,数据部分有若干填充字节,数据部分的最后一个字节用来表示填充的字节数;扩展 X 字段为

1b,若 $X=1$,表示此 RTP 首部后面还有扩展首部(较少使用);参与源数 s_num 字段为 4b,指出后面 CSRC 的个数;标记 M 字段为 1b,$M=1$ 表示该 RTP 报文具有特殊意义,例如在传送视频流时用来表示每一帧的开始。

载荷类型(payload type)字段占 7b,指出 RTP 数据所属的应用格式,收到 RTP 报文的应用层就是根据该字段指出的类型进行相应处理。如对于音频载荷,μ 律 PCM(0)、GSM(3)、LPC(7)、A 律 PCM(8)、G. 722(9)、G. 728(15)等;对于视频载荷,JPEG(26)、H. 261(31)、MPEG1(32)、MPEG2(33)等。

序号(sequence)字段为 16b,对每个 RTP 报文进行编号,每发出一个后序号加 1,与媒体流无关。在一次 RTP 会话开始时,初始序号可以随机选择。接收端依据序号对 RTP 报文进行排序,并发现报文丢失现象。

时间戳(time stamp)字段为 32b,反映了 RTP 报文中的承载数据的第一个字节的采样时刻。在一次会话开始时,时间戳初始值是随机选取的。即使没有信号数据发送,时间戳的数值也要随着时间流动而不断增加,使接收端可以据此同步媒体流的播放。在 RTP 中没有规定时间戳的粒度(granularity),而是取决于有效载荷的类型,因此 RTP 时间戳又称为媒体时间戳,以强调时间戳与信号类型的从属关系。例如,对于 8kHz 的 PCM 采样话音信号,若每隔 20ms 构成一个数据块,则一个数据块中应包含 160 个样本($0.02\times8K=160$),则发送端每发送一个 RTP 报文时间戳的值就增加 160。所以,时间戳字段记录的不一定是几分几秒的内容。

同步源标识符(Synchronous Source Identifier,SSRC)和参与源标识符(Contributing Source Identifier,CSRC)各为 32b,用以表示 RTP 流的来源。

SSRC 与 IP 地址无关,在新的 RTP 流开始时随机产生,可以有多个不同的流同时复用 UDP 进行传送,例如用不同角度的摄像机拍摄同一场景所产生的多个 RTP 流。接收端 UDP 将不同的 RTP 流送到各自的终点,分别进行处理。若发生两个不同的 RTP 流选择了同一个 SSRC 的情况(虽然概率很小),则两个源均应重新选择另一个 SSRC,如图 6.16 所示。

CSRC 是可选的,最多可有 15 个。SSRC 用以标识来源于不同地点的 RTP 流。在组播环境中,可以用中间的转发站(称为混合站 Mixer)将多个发往同一地点的 RTP 流混合成一个流,以节省通信资源,在目的地站点根据 CSRC 的数值将不同的 RTP 流分开。

实时传送控制协议(RTP Control Protocol,RTCP)是配合 RTP 执行控制操作的协议,是 RTP 不可或缺的组成部分(RFC 1889/1990)。

RTP 的主要功能是 QoS 的监视与反馈、媒体间的同步(视频和声音之间)、组播组中成员的标识。RTCP 报文也用 UDP 传输,但 RTCP 不负责媒体流的封装。由于 RTCP 报文很短,可以将多个 RTCP 报文组合承载在一个 UDP 数据报中。

RTCP 报文周期性地发送,携带发送端和接收端对 QoS 的统计信息报告,如已发送的报文数、字节数、报文丢失率、报文到达时间间隔的抖动等。

RTCP 有 5 种报文类型,使用同样的报文格式。

(1) 源点描述报文 SDES(类型 200)对会话中的参加者进行描述,包含参加者的规范名(Canonical Name,CNAME),一般是参加者的 E-mail 地址字符串。

(2) 结束报文 BYE(类型 203)用以关闭一个媒体流。

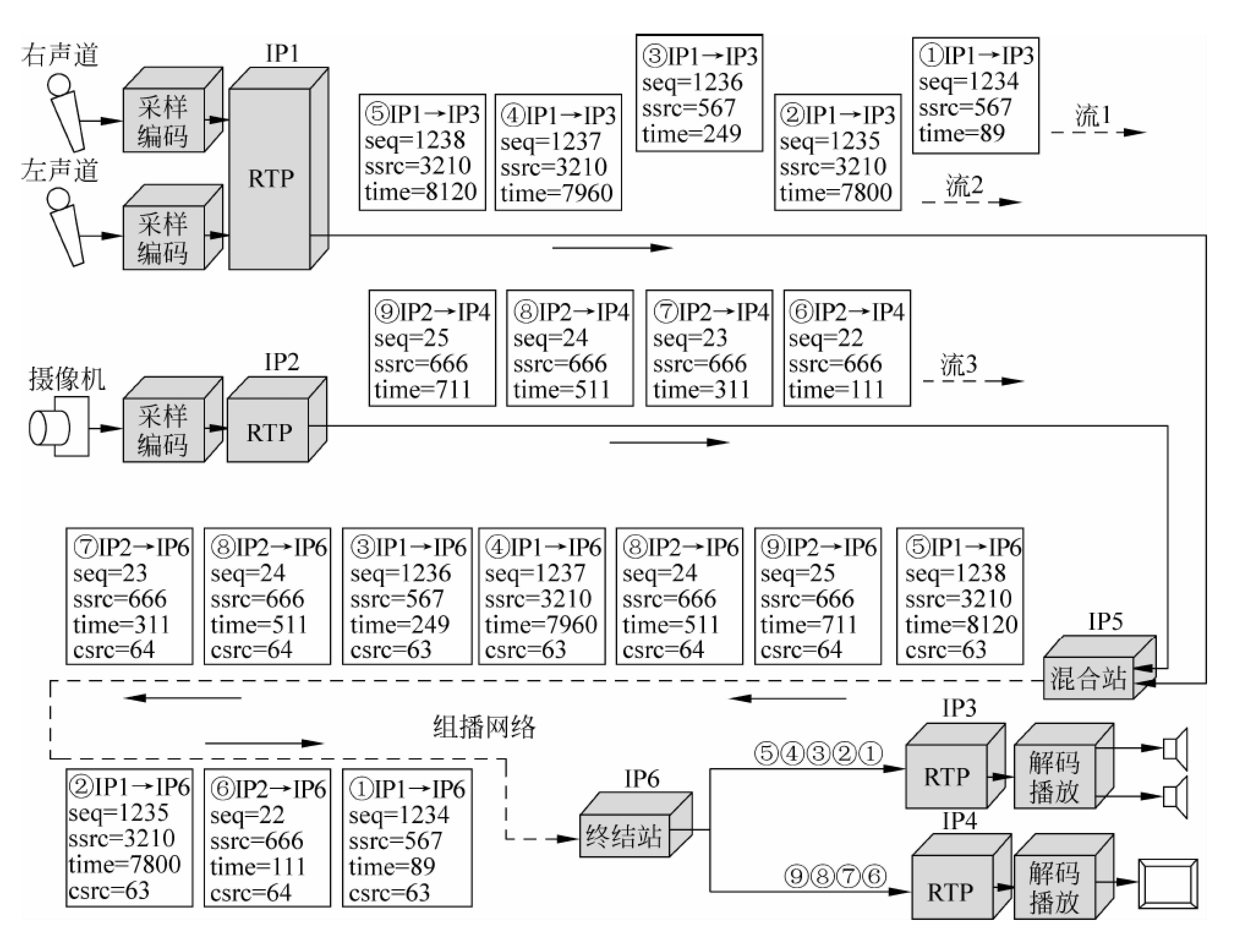


图 6.16　RTP 传输媒体流和报文示例

(3) 特定应用报文 APP(类型 204)使应用程序能够定义新的报文类型。

(4) 发送端报告报文 SR(类型 200)用以发送端周期性地向所有接收端进行报告,报文发送为组播方式。发送端每发送一个 RTP 流,就要发送一个 SR。SR 报文的主要内容包括:该 RTP 流的 SSRC;该 RTP 流中最新产生的 RTP 报文的时间戳和绝对时钟时间(墙上时钟时间即可,也即服务器系统时间);该 RTP 流包含的报文数;该 RTP 流包含的字节数。绝对时钟时间用于视频和声音的同步,因为属于不同的流,需要同一个参考时钟。

(5) 接收端报告报文 RR(类型 201)有两个目的:第一,可以使所有接收端和发送端了解当前网络的状态;第二,可以使所有发送 RTCP 报文的站点自适应地调整字节的发送 RTCP 报文的速率,使得用于控制的 RTCP 报文不要过多地占用 RTP 报文的带宽资源。RR 由接收端周期性地以组播方式向所有成员发送。接收端每增加一个 RTP 流(一次会话可能包含多个 RTP 流)就产生一个 RR。RR 报文的内容包括:所收到的 RTP 流的 SSRC;该 RTP 流的报文丢失率(可用于发送端流控);在该 RTP 流中的最后一个 RTP 报文的序号;报文到达时间间隔的抖动等。一般要求 RTCP 报文的通信量不能超过网络中数据报文通信量的 5%,而 RR 的通信量应小于所有 RTCP 通信量的 75%。

实时流式协议(Real-Time Streaming Protocol,RTSP)是配合 RTP 工作的另一种控制协议,由 RFC 2326 定义。

RTSP 以 C/S 方式工作,用于多媒体播放控制。用户通过 RTSP 提供的服务,在播放器客户端软件中播放流媒体时,可以进行各种时移操作,如暂停/继续、后退、快进等,就像控制录像机一样,因此,RTSP 又被形象地称做互联网录像机遥控协议。

如图 6.17 所示，RTSP 与 RTP 和 RTCP 各司其职，RTP 用来封装和传输媒体流数据，RTCP 用来保障 QoS，而 RTSP 用来协调客户机和服务器，能够实现媒体流的传送控制。因此 RTCP 和 RTSP 可视为 RTP 的信令协议，在物理信道上是带内的，在协议处理上是带外的。RTSP 的语法与 HTTP 相似，RTSP 控制报文既可以在 TCP 上传送，也可在 UDP 上传送。

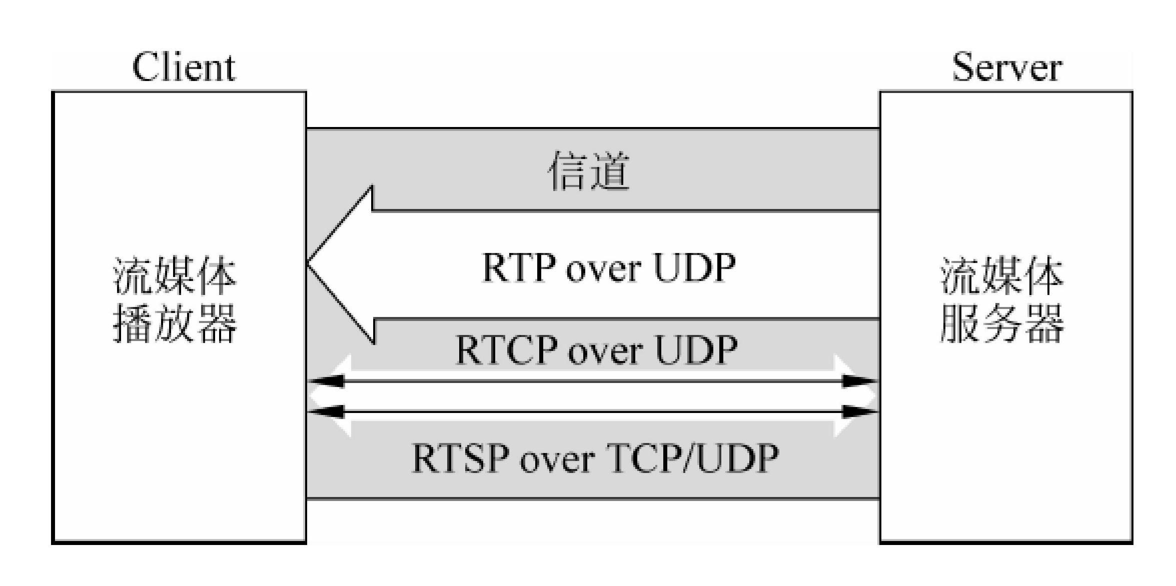


图 6.17　RTSP 与 RTP 和 RTCP 的关系

6.3.3　SIP

会话发起协议（Session Initiation Protocol，SIP）是一种应用层控制协议（RFC 2543），属于纯文本型的信令协议，采用 HTTP 的首部、编码规则等许多内容，可以管理不同接入网络上的会话，包括终端设备之间任何类型的通信，如网络电话、视频会议、即时消息处理或协作会议。SIP 对业务不作定义或限制，传输、QoS、计费、安全性等问题都由基本核心网络和其他协议处理。

如图 6.18 所示，SIP 系统有两种构件：用户代理（user agent）和网络服务器（network server）。用户代理由两个部分组成：用户代理客户机（user agent client）和用户代理服务器（user agent server），前者用以发起呼叫，后者用以接收呼叫。网络服务器分为代理服务器（proxy server）和重定向服务器（redirect server），前者接收来自主叫用户的呼叫请求，并转发给下一跳代理服务器，最后将呼叫请求转给被叫用户，后者通过响应（如果需要的话）告诉客户机下一跳代理服务器的地址，由客户按此地址向下一跳代理服务器重新发送呼叫请求。SIP 还通过会话描述协议（Session Description Protocol，SDP）维护电话会议参加者的动态加入或退出。

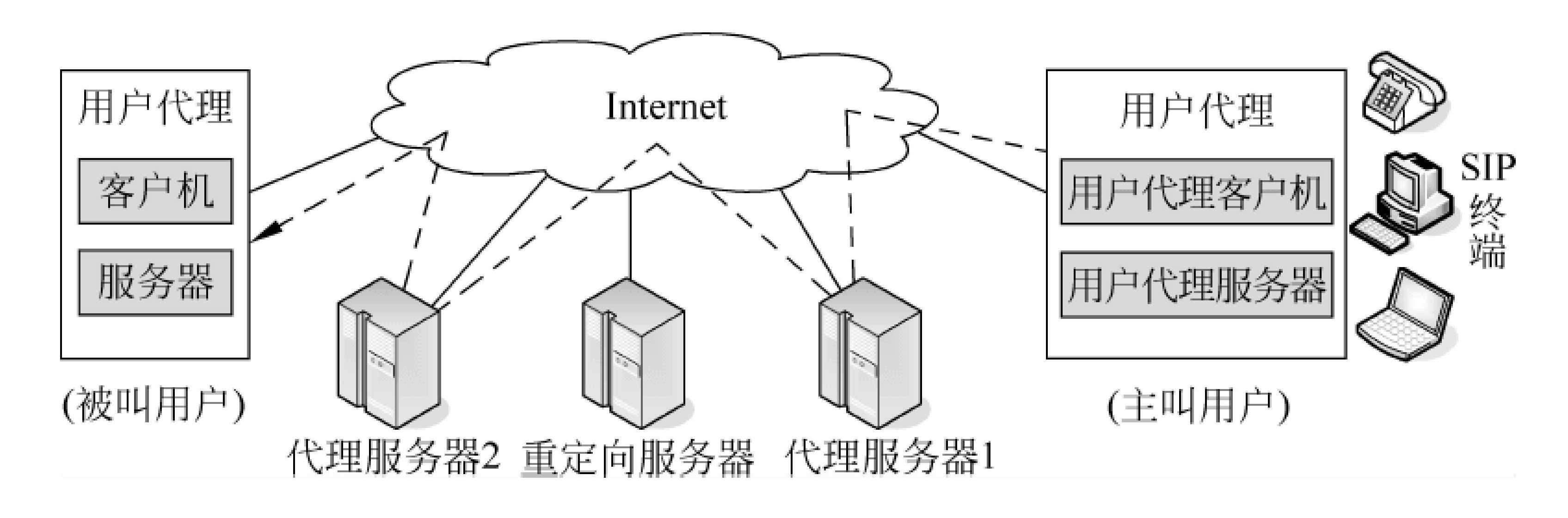


图 6.18　SIP 技术组网结构

SIP 和 MGCP、H.323、H.248 都是 VoIP 应用协议，分别是电信网（标准化组织 ITU）与互联网（标准化组织 IETF）两大阵营推出的标准。H.323 和 H.248 试图把 IP 电话当做

众所周知的传统电话,只不过传输方式发生了改变,由电路交换变成了分组交换;而 SIP 和 MGCP 的目标是将 IP 电话业务变成互联网上的一个成功的应用。

ITU-T H.323 采用基于 ASN.1 和压缩编码规则的二进制方法表示其消息,制定了在无 QoS 保证的分组网络(Packet-Based Networks,PBN)上的多媒体通信系统标准。如图 6.19 所示,H.323(阴影部分)是一个框架性标准,涉及终端设备、视频、音频和数据传输、通信控制、网络接口方面的内容,还包括组成多点会议的多点控制单元(Multi-point Control Unit,MCU)、多点控制器(Multi-point Controller,MC)、多点处理器(Multi-point Processor,MP)、网关(Gateway,GW)以及网守(GateKeeper,GK)等设备(如图 6.20 所示)。

<table>
<tr><td colspan="2">音频视频应用</td><td colspan="4">信令和控制</td></tr>
<tr><td>音频
编解码</td><td>视频
编解码</td><td rowspan="2">RTCP</td><td rowspan="2">H.225.0
注册
信令</td><td rowspan="2">H.225.0
呼叫
信令</td><td rowspan="2">H.245
控制
信令</td></tr>
<tr><td colspan="2">RTP</td></tr>
<tr><td colspan="4">UDP</td><td colspan="2">TCP</td></tr>
<tr><td colspan="6">IP</td></tr>
</table>

图 6.19 H.323 协议结构

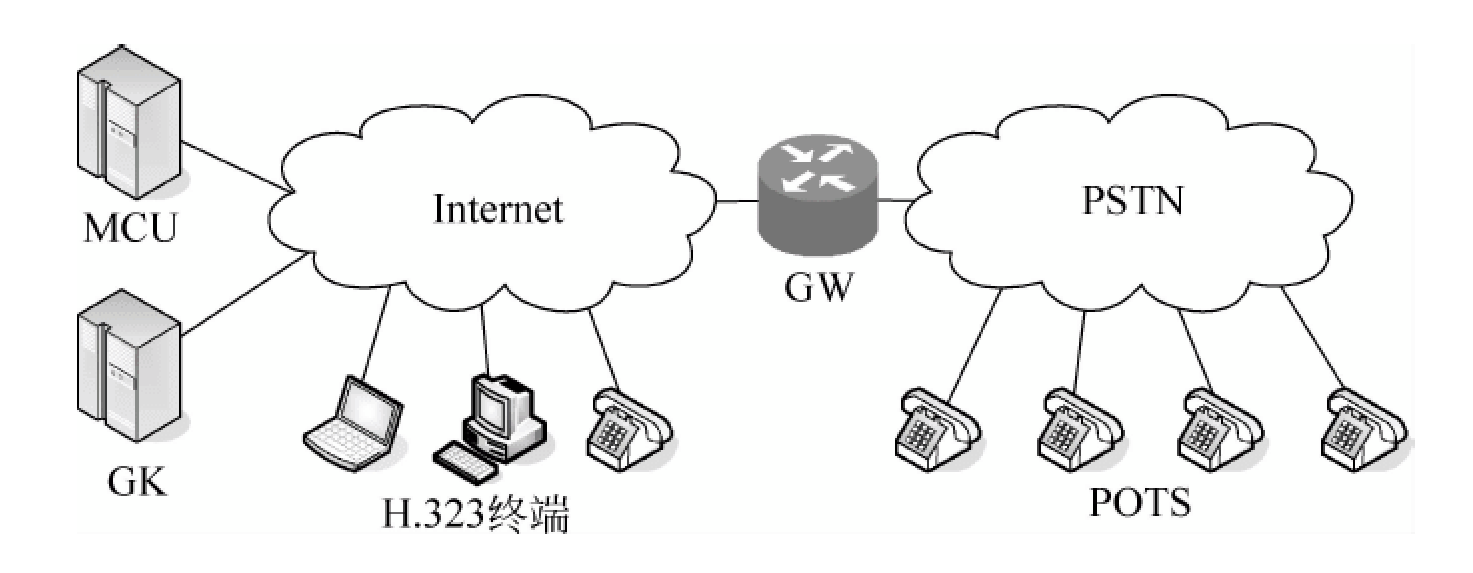


图 6.20 H.323 技术组网结构

传统的电信网络可提供高质量的通信服务,是与其可运营的思想、技术体制和系统架构分不开的。电信网络以语音业务为中心,而不像计算机网络那样以数据为中心。那么,在新一代 IP 网络上理应继承这一优势,为多媒体应用的开展创造良好的条件。

与电信网络的核心设备电话交换机(Telephone Exchanger)一样,新一代 IP 网络上的 VoIP(Voice over IP)业务采用**软交换机**(Soft Switch)。

软交换机是基于分组网络、利用程控软件提供呼叫控制功能的设备和系统。但软交换机本身并没有整体组网技术机制和网络体系架构,而需要依赖分组网络(IP、ATM 等)和相关设备来实现。

软交换机通过媒体控制协议技术可以实现呼叫控制与媒体传输相分离的思想。所以,软交换机也称为呼叫代理器(Calling Agent,CA)或媒体网关控制器(Media Gateway Controller,MGC)。软交换机概念的提出不仅使基于 IP 的语音业务功能和与传统 PSTN 网的交换机功能可以完全兼容,还能够从根本上确保 IP 电话技术能够完全替代传统电信网络电话服务。

软交换机的技术基础是媒体网关控制协议(Media Gateway Control Protocol,MGCP),

ITU-T 则定义为 H.248 标准。MGCP/H.248 又称器件控制协议，是一种具有主从关系的控制协议。被控制方一般为非智能的简单器件或设备，一切状态、事件的发生和变化都必须上报主控设备。所以，媒体控制协议不应被视为呼叫信令，只能用于端点（包括用户端点和中继端点）的控制。软交换机的局间呼叫信令协议为 H.323/SIP。

下一代网络（Next Generation Network，NGN）是对传统电信网络的升级换代，采用 IPv6 为互连平台，完全进化成为计算机网络。但 NGN 技术体系架构上却存在两种不同的策略和思路。

一种思路强调智能的端点和边缘、简单的网络设施。因为端到端多媒体融合业务的驱动力来源于端点和边缘，下一代网络业务的创新和发展正依赖于此。互联网的成功也证明了这种思路对多媒体业务的重要性。H.323 和 SIP 及其相关的网守和代理服务器正是这一技术体制的体现。

另一种思路主张简单的端点和边缘、智能的网络设施。因为只有单纯统一的端点和边缘设备，才有利于规模性地经营管理和控制。传统 PSTN 所提供的语音业务已验证了这种思路在实际应用中的有效性。MGCP/H.248 协议及软交换机正是体现这一技术体制的实现手段。

然而，NGN 实际上既需要智能端点业务的技术特性，又需要实现规模化经营管理。难点在于如何综合考虑多种技术以达到技术和运营的平衡。实际上，相互间的融合才是建设 NGN 及其业务的最佳途径。

6.4　宽带网络接入协议

6.4.1　PPP

点对点协议（Point-to-Point Protocol，PPP）是一种面向字符型的数据链路层协议（RFC 1661～1663），用于在异步串行通信线路（也适用于同步串行线路）上传输分组型协议（如 IP），所以在拨号上网业务中得到广泛应用。PPP 与 SLIP 协议在工作原理和应用目标上都很类似，但相比 SLIP 的简单性，PPP 具备更强的功能、更好的性能，包括差错检测、协商配置和鉴别协议、分配临时 IP 地址、支持更多网络层协议等。

如图 6.21 所示，PPP 可把 IP 报文封装到串行线路上进行传输，还包括一个链路控制协议（Link Control Protocol，LCP）、一套网络控制协议（Network Control Protocol，NCP）。LCP 定义了 11 种类型的帧，用以在数据链路层上建立、配置和测试连接关系，并提供给通信双方协商选项的方法；NCP 为网络层的一个子层，用于支持不同的网络层协议的参数配置，例如对 IP 而言，NCP 为 IPCP，可用于配置 IP 地址。

PPP 和 LAP/HDLC 帧格式相似，如图 6.22 所示，但 PPP 是面向字符的，而 LAP/HDLC 是面向比特的。

2 字节的协议字段表示 PPP 帧数据字段中的报文类型，如 0x0021（IP）、0xc021（LCP）、0x8021（NCP）、0xc023（PAP）、0xc223（CHAP）。其中 PAP 和 CHAP 为安全认证（鉴权）协议。

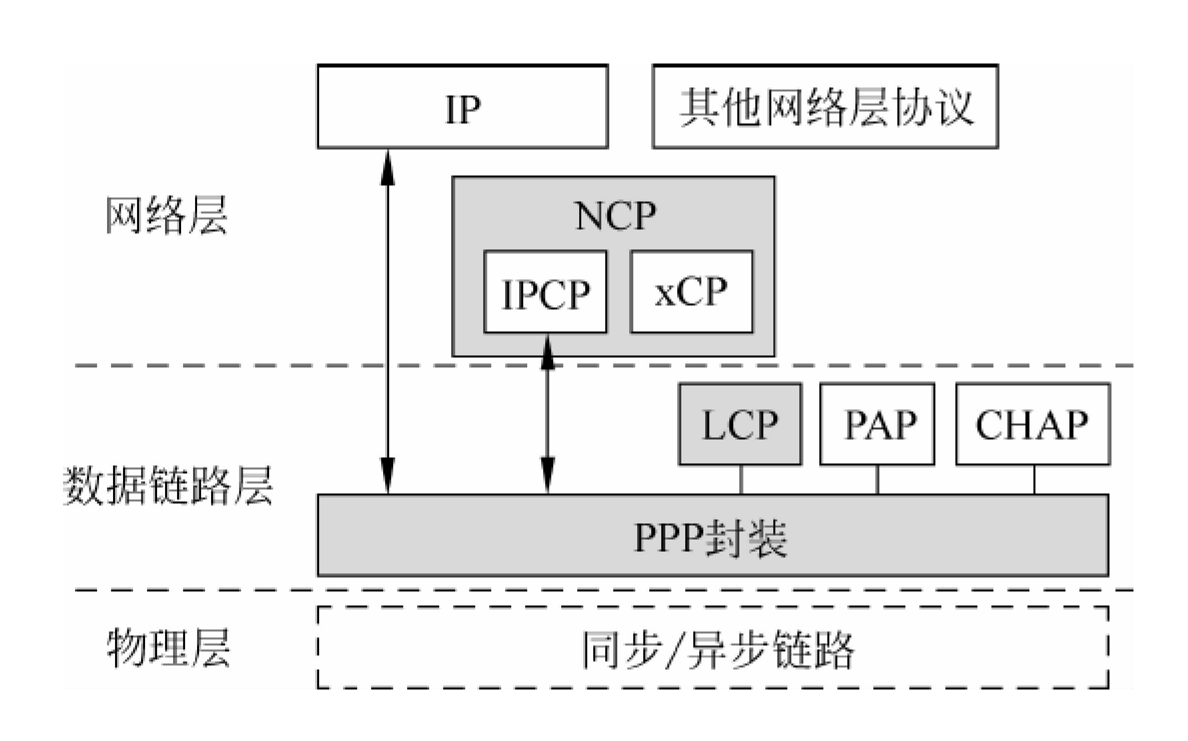
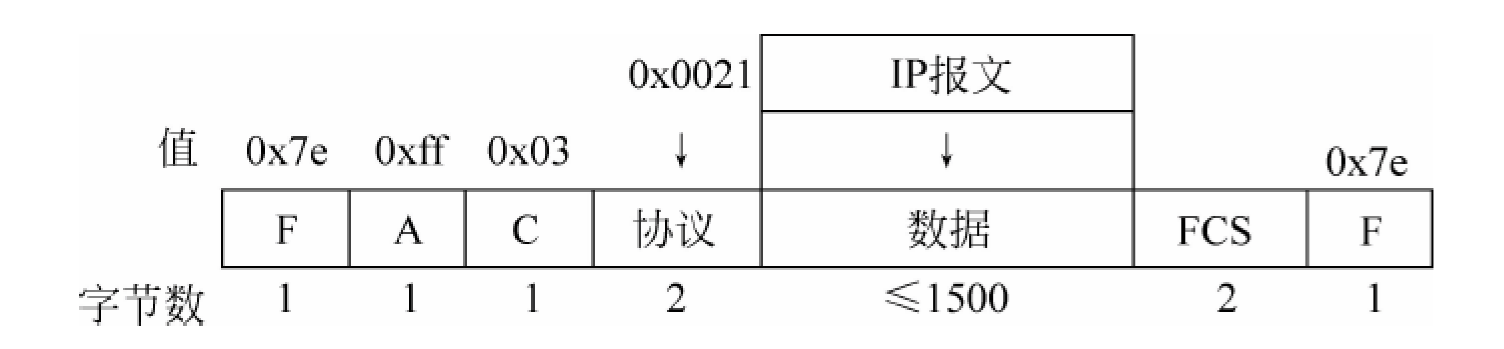


图 6.21 PPP 协议栈



值	0x7e	0xff	0x03	0x0021	IP报文		0x7e
	F	A	C	协议	数据	FCS	F
字节数	1	1	1	2	≤1500	2	1

图 6.22 PPP 帧格式

由于 PPP 为字符型协议,以 0x7e 为帧同步码(flag),为了避免数据字段出现的 0x7e 编码引起误解,PPP 也采用类似 SLIP 的报文透明传输方法,转义字符为 0x7d,转换规则为:在数据字段中,0x7e→0x7d-0x5e,0x7d→0x7d-0x5d。

图 6.23 所示为 PPP 的工作状态和转换关系。

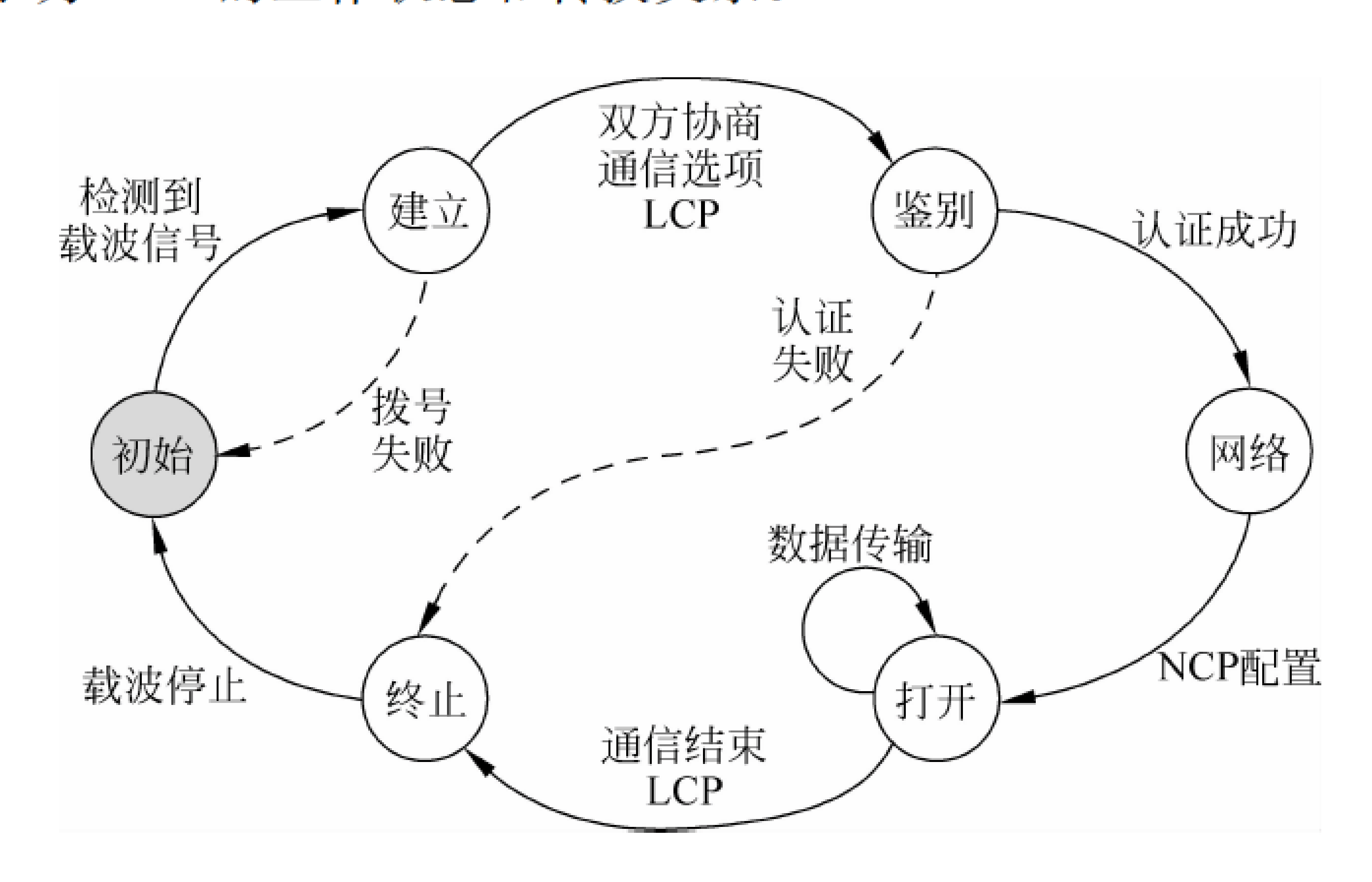


图 6.23 PPP 工作状态及其转换关系

当用户接入 ISP 时,首先拨号建立物理连接;然后,如图 6.24 所示,由 LCP 协商将要使用的 PPP 参数(最大帧长、鉴权协议、报头压缩等);若随后的 PAP/CHAP 鉴权成功,将使用 NCP 给接入的计算机分配 IP 地址,完成接入网络的过程;当用户上网完毕,释放 NCP 和 LCP 连接,断开拨号线路。

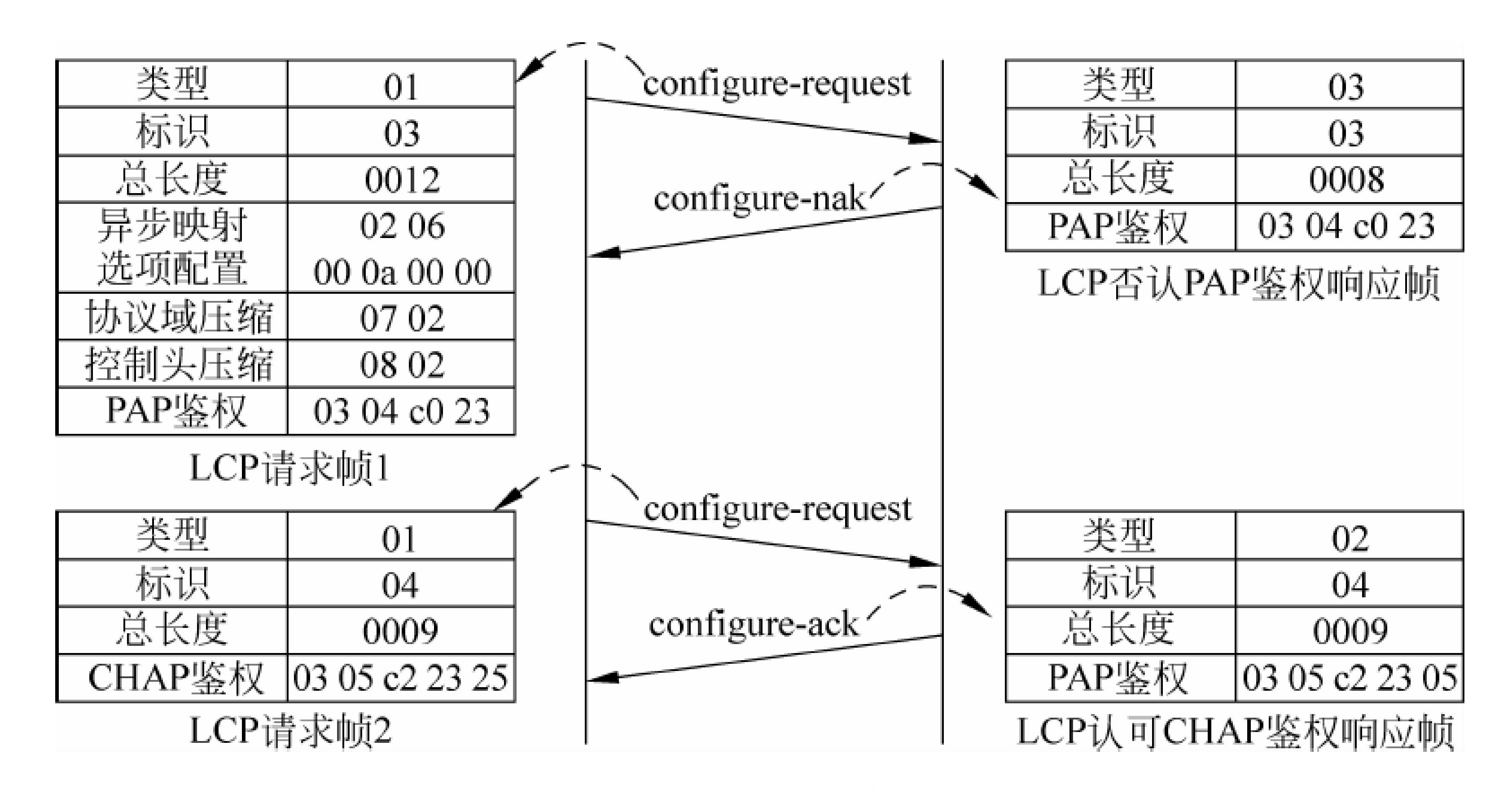


图 6.24　LCP 协商参数示例

6.4.2　PPPoE

在传统 PSTN 拨号上网中，采用异步通信端口上的 PPP 或 SLIP 承载 IP 报文，实现面向字符到面向分组(面向比特流)的通信方式转换。但在 ADSL 等宽带接入业务中，计算机设备通过 Ethernet 接口连接 ADSL-Modem，不适合直接运行 PPP。此外，如图 6.25 所示，从 ADSL 技术架构中可以看出，ADSL 不需要拨号穿越 PSTN，而是一种用户环路构成的点对点连接关系。

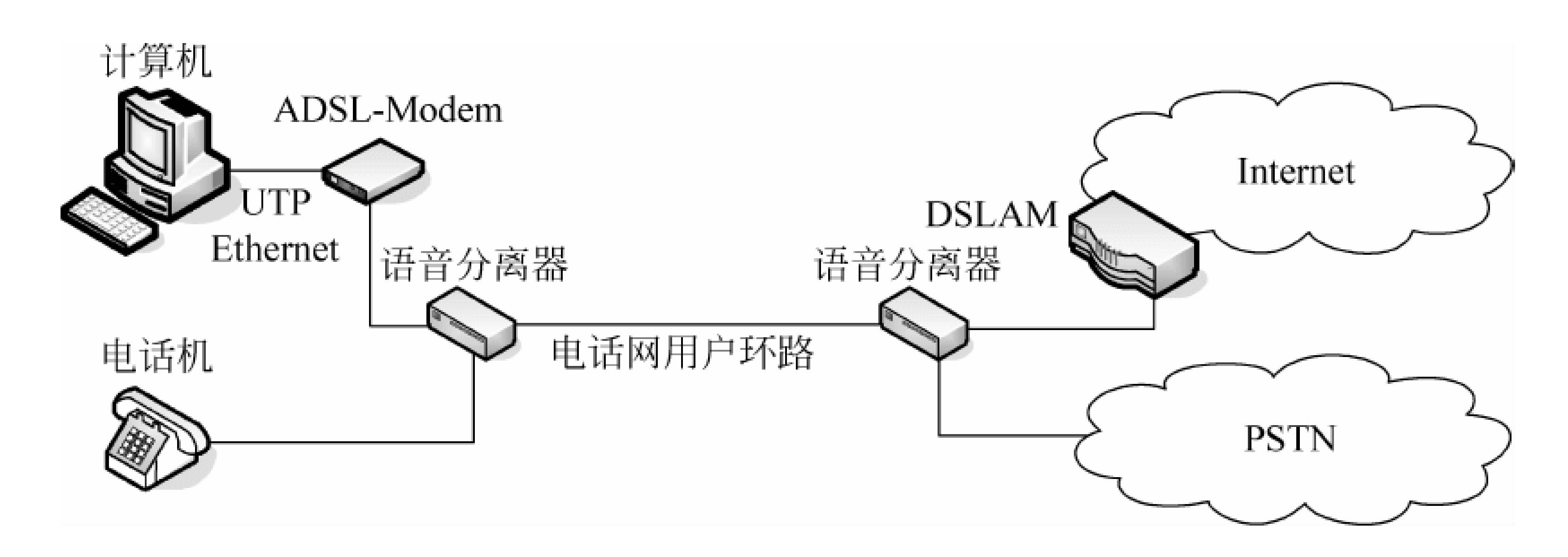


图 6.25　ADSL 用户-网络接口示意

宽带网络接入采用 PPPoE(PPP over Ethernet)协议(RFC 2516)，引入虚拟拨号(virtual dial-up)的概念，与 PPP 一起实现上网认证、地址分配等管理功能。

如图 6.26 所示，PPPoE 运行在 Ethernet 协议上，并承载 PPP。

PPPoE 协议执行过程分发现(discovery)和会话(session)两个阶段。

发现阶段用于交换设备的信息、协商建立 PPPoE 会话，code 为类型码，协议类别 Ether_type＝0x8863，payload 承载的是 PPPoE 的 TAG，即控制报文。TAG 格式为：＜tag-type(16b)＞ ＜tag-length(16b)＞ ＜tag-value＞。payload 中可包含多个 TAG。发现阶段是无状态的，由计算机终端和局端通信服务器(又称接入集中器)相互获取 MAC 地址等信息，并确定本次会话的 session_id 值。协议执行分以下四个步骤。

(1) 终端发送 PPPoE 主动发现发起(PADI)报文，目的 MAC 地址为 Ethernet 广播地址(全 1)。其中 code＝0x09，session_id＝0x0000。PADI 报文必须至少包含一个服务名称类型的标签(tag-type＝0x0101，意为 service-name)，向通信服务器提出所要求提供的服务。

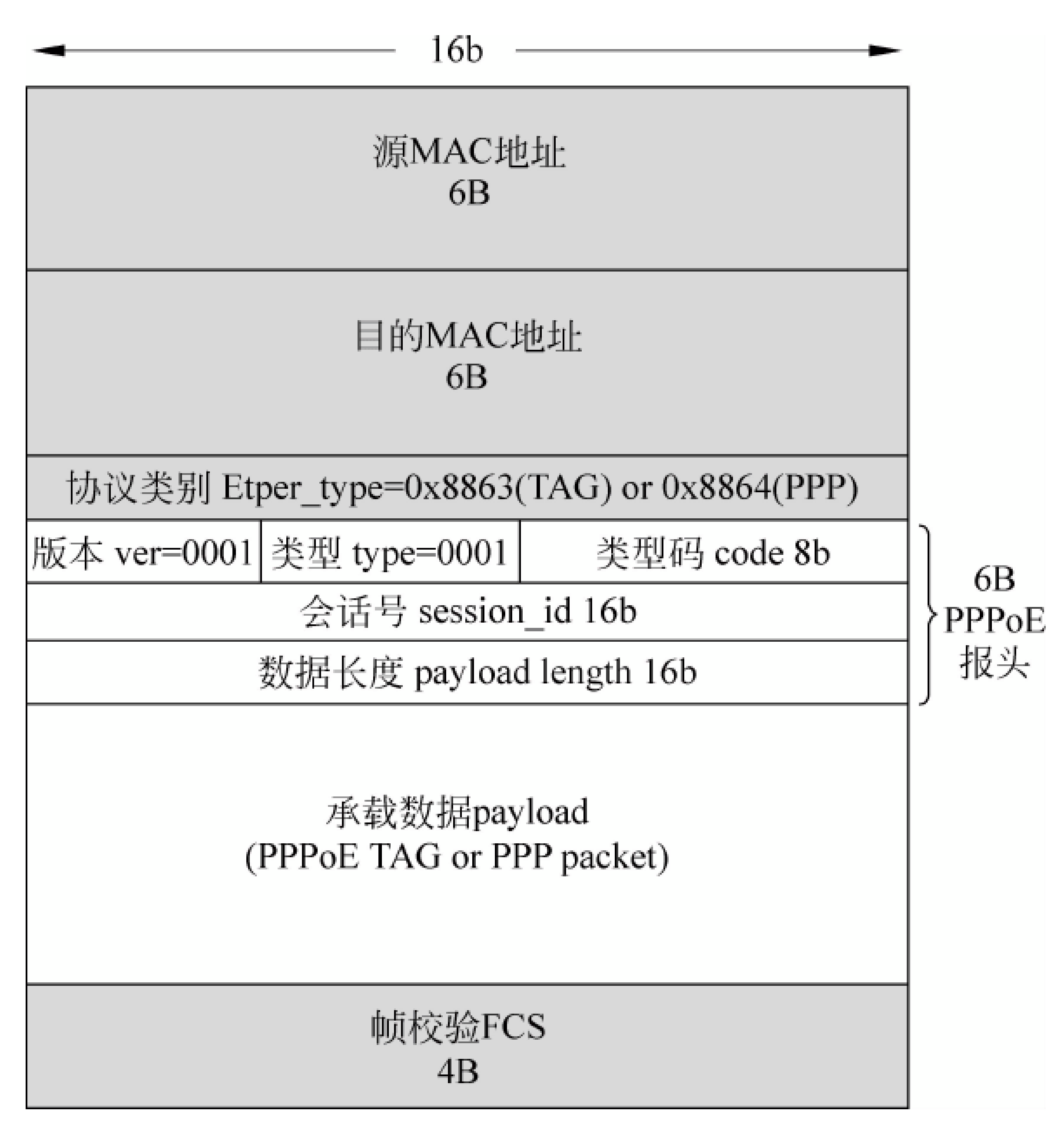


图 6.26 PPPoE 报文格式

(2) 通信服务器收到服务范围内的 PADI 报文,发送 PPPoE 主动发现提供(PADO)报文,以响应请求。其中 code=0x07,session_id=0x0000。PADO 分组包含通信服务器名称类型的标签(tag-type=0x0102,意为 ac-name)以及一个或多个服务名称类型标签,表明可向终端提供的服务种类。

(3) 终端在可能收到的多个 PADO 报文中选择一个,然后向所选择的通信服务器发送 PPPoE 主动发现请求(PADR)报文。其中 code=0x19,session_id=0x0000。PADR 报文必须包含一个服务名称类型标签,确定向通信服务器(或交换机)请求的服务种类。

(4) 通信服务器收到 PADR 报文后,准备开始 PPP 会话,发送 PPPoE 主动发现会话确认(PADS)报文。其中 code=0x65,session_id 字段值为通信服务器生成的唯一的 PPPoE 会话标识号码。PADS 报文也必须包含一个通信服务器名称类型的标签以确认向终端提供的服务。当终端收到 PADS 报文获得确认后,双方就进入 PPP 会话阶段。

在会话阶段,Ethernet 报文均为单播发送,协议类别 Ether_type=0x8864。PPPoE 的 code=0,payload 只封装和传输 PPP 报文,进一步由 PPP 实现协商(LCP)、鉴权(PAP/CHAP)、分配 IP 地址(NCP)、IP 报文传输等功能,session_id 在会话阶段保持发现阶段获得的值不变。

封装的 PPP 报文从 2B 的协议标识码开始。当传输 IP 报文时,考虑到 PPPoE 和 PPP 共占用 6+2=8B 报头,而 Ethernet 的 MTU 为 1500B,因此 IP 报文的最大长度为 1492B。

PPPoE 还有一个主动发现终止(PADT)报文,可以在会话建立后的任何时候发送,用于终止 PPPoE 会话。PADT 可以由终端或通信服务器发送。当对方接收到一个 PADT 报文,就不再允许使用这个会话来发送 PPP 报文。PADT 分组不需要任何标签,其 code=0xa7,session-id 字段值为需要终止的 PPP 会话标识号。正常情况下应该使用 PPP 来终止 PPPoE 会话,但是当 PPP 不能使用时,可以使用 PADT 报文。

6.4.3 MPCP

FTTx(Fiber To The x)是对光纤接入网一系列技术的统称,光纤到户(FTTH)是宽带网络接入的理想目标。FTTx 有别于 HFC 的模拟光纤主干网,而是全数字化的计算机网络技术。比较常见的 FTTx 方式有以下几种。

(1) FTTC(FTT-Curb)光纤到路边。

(2) FTTZ(FTT-Zone)光纤到小区。

(3) FTTB(FTT-Building)光纤到大楼。

(4) FTTF(FTT-Floor)光纤到楼层。

(5) FTTH(FTT-Home)光纤到家庭。

以常见的 FTTB 为例,一般采用 FTTB+LAN 的模型。接入网的主干光纤网络进入大楼(或园区),再通过 10/100Mb/s 以太网交换机(或集线器)组网并互连各个用户,用户认证、地址分配、计量计费等管理通过配置相应的接入控制服务器完成。因此,用户只需使用已有的以太网网卡和 UTP5 双绞线即可接入 Internet,不需要额外的传输设备(如 Modem 或 STP)。

FTTH 则将光纤延伸到接入用户身边,具有高带宽、高可靠性、高安全性、高扩展性的优势,但是,传统的有源光交换和光传输设备相当昂贵、易受环境影响、难以部署、维护困难,不利于普及应用,所以,稳定、灵活的无源光传输技术和设备对 FTTH 的实现十分关键。

无源光网(Passive Optical Network,PON)采用无源器件构造最后一公里的光纤接入线路,光进铜退、光纤入户,配合骨干网、汇聚网的光交换技术,可形成真正的全光网络(All Optical Network)。

ITU-T G.982 提出了光纤接入网(Optical Access Network,OAN)功能参考配置模型,如图 6.27 所示。

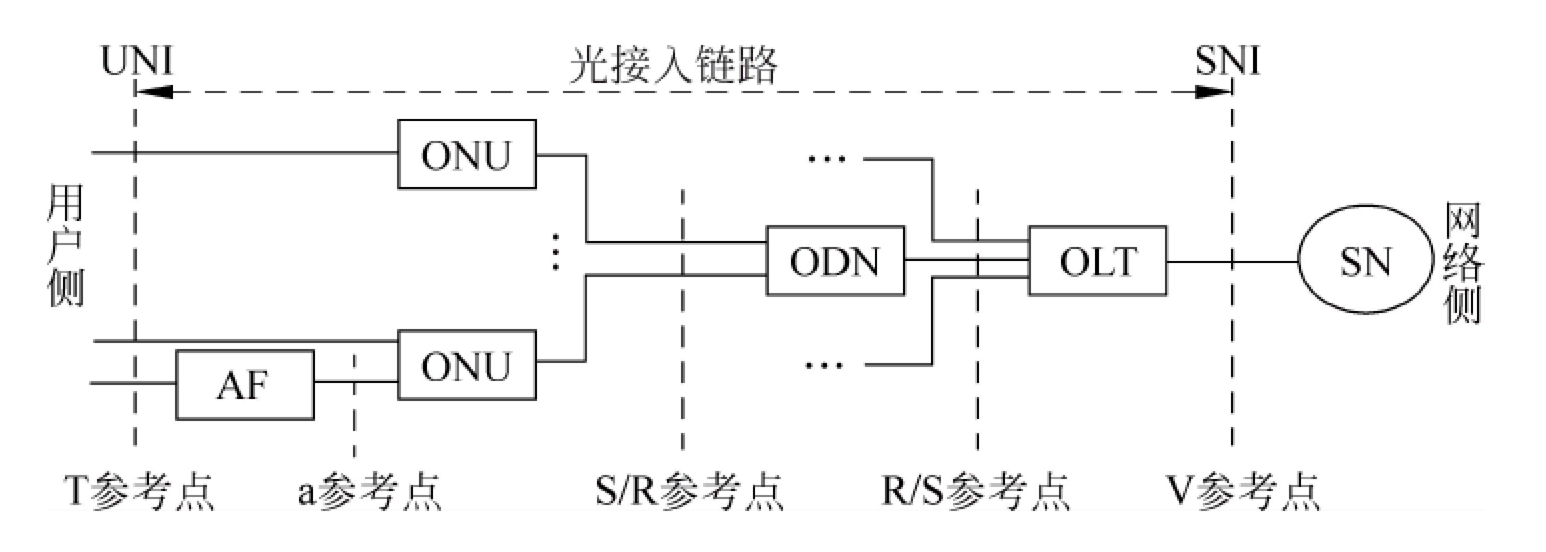


图 6.27　OAN 参考模型

(1) SN 为光交换网络(Switching Network),由光交换机构成,是接入网或汇聚网的一部分。

(2) OLT 为光线路终端(Optical Line Terminal),也称局用数字终端(HDT),与 SN 接口,分离交换和非交互业务,经一个或多个 ODN 与用户侧 ONU 通信,实现网络管理的主要功能,管理来自 ONU 的信令和监控信息,为 ONU 和本身提供维护功能。

(3) ODN 为光配线网(Optical Distribution Network),实现光信号功率分配,由光连接器、光分路器等无源光器件组成,使多路 ONU 共享光传输信道和光器件。

(4) ONU 为光网络单元(Optical Network Unit),通过光电、模数转换、复用、信令处理和维护管理,提供直接的或远端的用户电接口。

(5) AF 为适配功能(Adaptation Function),用于 ONU 和用户设备间的接口适配,可以集成在 ONU 中。

PON 采用称为无源光分路器(splitter)的 ODN 设备将 OLT 信号从 OLT 分送到用户端 ONU。下行为 TDM 广播式发送,上行通常采用时分复用多址访问(TDMA)完成共享信道信息传输(如图 6.28 所示)。

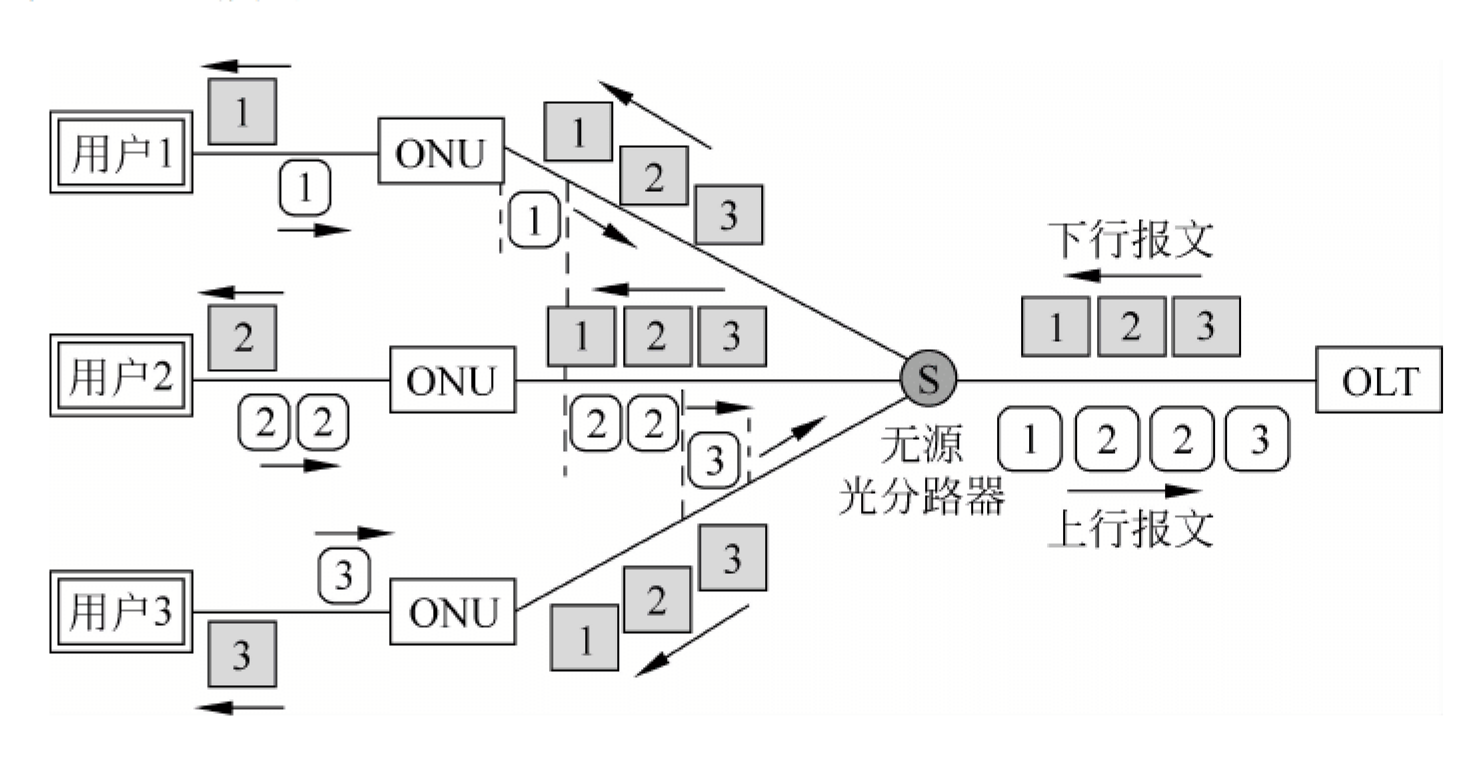


图 6.28 PON 工作原理

PON 技术包括基于 ATM 的 PON(APON,ITU-T G.983)、基于 Ethernet 的 PON (EPON,IEEE 802.3-2005)、千兆 PON(GPON,ITU-T G.984)三类,差别是采用了不同的数据链路层技术。

由于 EPON 采用高速、易用、低价、普及的 Ethernet 技术,在宽带接入网中具有明显的比较优势。

EPON 物理层使用波分复用(Wave Division Multiplexing,WDM)技术,实现单纤双向传输,下行波长为 1490nm,上行波长为 1310nm,定义了 1000Base-PX-10U/D(支持最长 10km)和 1000Base-PX-20U/D(最长 20km)两种光接口。在物理编码子层,采用 8B/10B 线路编码,支持 1.25Gb/s(可升级到 10Gb/s)对称通信速率。

多点控制协议(Multi-Point Control Protocol,MPCP)是 EPON 关键技术之一,由以太网 MAC 协议扩展而来,增加了点到多点通信控制功能。

MPCP 实现 ONU 的发现和注册、多个 ONU 之间上行传输资源的分配、动态带宽分配、网络启动过程控制等。这些功能由允许选通(gate)、报告(report)、注册请求(register_req)、注册(register)和注册确认(register_ack)5 种 MAC 帧实现。例如最基本的用于请求和带宽分配的 gate/report 机制,OLT 通过 gate 给 ONU 授权,分配使用间隙的长短、允许其发送数据的开始时刻,而 ONU 通过 report 报告自身状态,包括需要发送数据的尺寸、本地时钟同步等信息。

为避免上行信号相互重叠,MPCP 要求 OLT 与 ONU 的时间保持严格同步。OLT 和 ONU 均有一个本地计时器,以 16ns 为单位,作为时间戳在发送时填入 MPCP 帧中。ONU 利用 OLT 发来的控制帧中的时间戳动态调整本地计时器的值,与 OLT 同步。但距离、光电器件性能、环境条件等差别导致各个 ONU 和 OLT 之间信息往返时延(Round Trip Time,RTT)可能差异很大,需要把每个 ONU 都调整到与 OLT 有相同的逻辑距离,即延迟时间均为 T_d。为达到这一要求,各个 ONU 应把上行帧延迟 $T_d-\text{RTT}_i$ 的均衡延迟时间再发送。

RTT 的获得采用时间标签测距法。如图 6.29 所示,要求 ONU 一旦收到 OLT 发送的 MPCP 帧,必须用帧所携带的时间戳更新本地计时器。这样,当 OLT 接收到 ONU 发送的帧时,就可利用时间戳与本地计时器之差计算出 RTT。

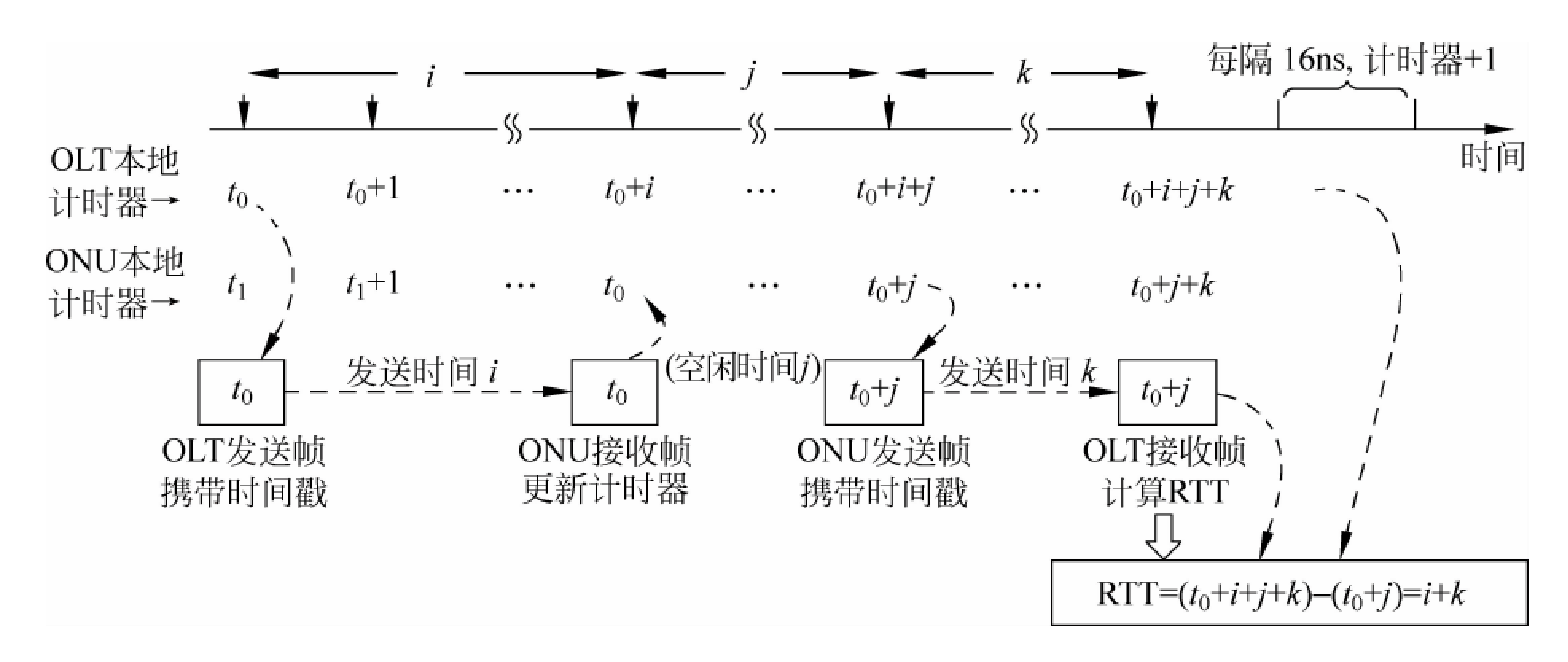


图 6.29　MPCP 时间标签测距法

OLT 启动后,将周期性地广播允许接入的时隙等信息;ONU 上电后,根据 OLT 广播的允许接入信息,主动发起注册请求;OLT 通过对 ONU 的认证(可选过程),允许 ONU 接入,并给请求注册的 ONU 分配一个该 OLT 端口唯一的逻辑链路标识(Logical Link Identifier,LLID)。

MAC 帧前的 7B 同步码(0x55)与 1B 开始符(0xd5)被 802.3z 定义为前导码(preamble),可携带控制信息。LLID 即位于 8B 前导码的第六、第七字节。下行 TDM 方式广播被 ONU 接收后,ONU 只接受 LLID 属于自己的帧,其余帧被舍弃。

C H A P T E R 7

第7章 移动通信网络

7.1 移动通信网络结构

移动通信网络(Mobile Communication Network,MCN)是采用无线通信技术的电话网络。20世纪70年代推出的**第一代移动通信网络**(1st Generation MCN,1G)就是运用FDMA复用方式的电路交换模拟电话系统。

但移动通信并非无线通信与PSTN的简单叠加。作为面向语音通信的运营级网络,移动通信系统应支持完善的用户认证和管理、终端移动漫游、通话过程无缝切换等特定功能,支持多用户并发接入和通信,并提供全面的无线信号覆盖和稳定可靠的服务质量。

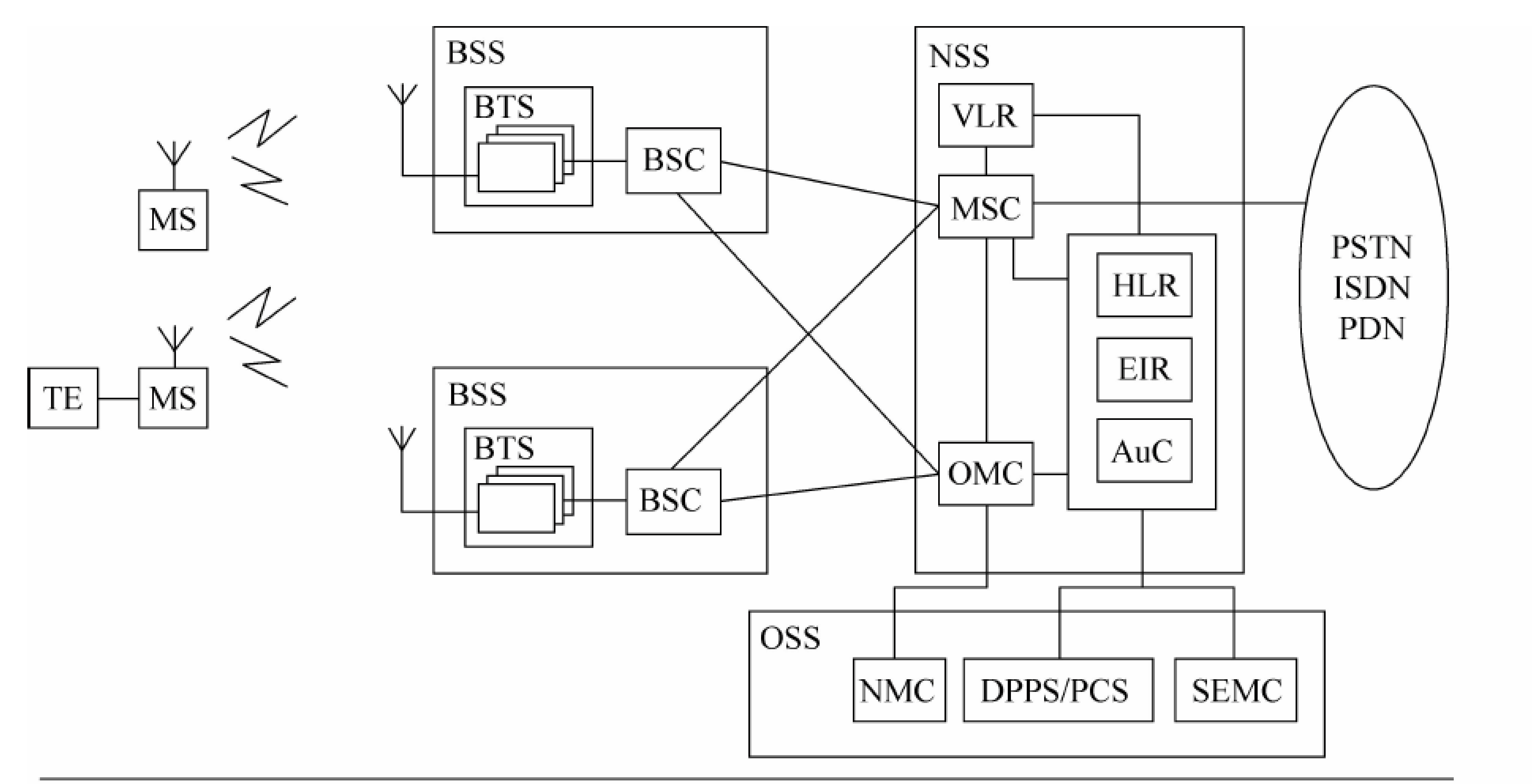
移动通信系统的体系结构如图7.1所示。

移动站(Mobile Station,MS)或移动主机(Mobile Host,MH)一般即指手机,通过无线基站(BSS)接入移动通信网络,呼叫管理、用户管理、系统管理、网络互连等功能由网络子系统(NSS)和运营支撑系统(OSS)协同完成。

由于适合移动通信的频谱资源非常有限,为能够充分利用资源为更多的用户提供更高质量的服务,需要通过多种复用技术来获得更大的容量。

移动通信系统首先采用**蜂窝**(cell)结构的复用方法,如图7.2所示,在地理空间上进行划分,理想情况下可以达到全面覆盖。一个蜂窝就是一个基站天线的覆盖范围,也称为小区,频率资源可以在不同的小区中重复使用。若干个小区构成一个**区群**,区群中各基站以有线通信方式连接到MSC。因此,移动通信网络常被称为**蜂窝通信网络**。

在蜂窝式移动通信系统中,一个移动终端从属于一个**家乡网络**(home network),由家乡地址相互区别;在家乡网络中,家乡代理跟踪移动终端的当前位置;当移动终端离开家乡网络到**外地网络**(foreign network)时,需要到外地代理上注册,并通过外地代理保持与家乡代理之间的联系,使家乡代理始终了解其位置。这是位置管理的基本工作原理。位置管理和切换管理一起构成移动性管理,是移动通信技术中最为独特和重要的内容。



MS: 移动台(Mobile Station)
TE: 终端设备 (Terminal Equipment)
BSS: 基站子系统 (Base Station Subsystem)
NSS: 网络子系统 (Network Subsystem)
OSS: 运营支撑系统 (Operation Surporting System)
OMC: 运营维护中心 (Operation Maintenance Centre)
NMC: 网络管理中心 (Network Management Centre)
AuC: 鉴权中心 (Authentication Centre)
PDN: 分组数据网 (Packet Data Network)
BTS: 基站收发信机(Base Transceiver Station)
BSC: 基站控制中心(Base Station Center)
VLR: 访问位置寄存器(Visitor Location Register)
MSC: 移动业务交换中心(Mobile Switching Center)
HLR: 归属位置寄存器(Home Location Register)
EIR: 设备识别寄存器(Equipment Identifier Register)
DPPS: 数据后处理系统(Data Post-Process System)
PCS: 用户识别卡个人化中心(Personalizatiom center)
SEMC: 安全性管理中心(Security Management Center)

图 7.1　移动通信系统体系结构

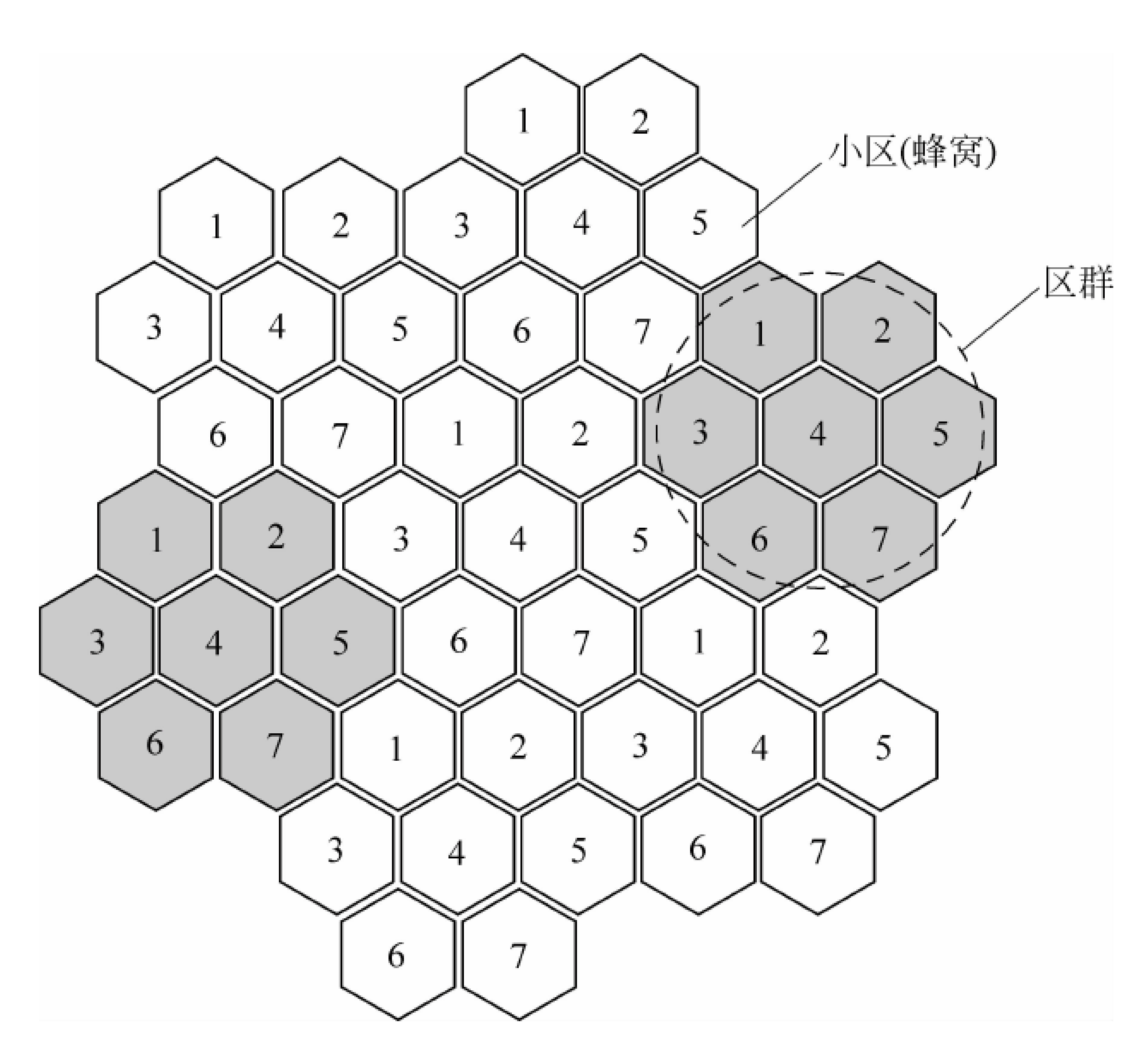


图 7.2　移动通信蜂窝示意

20 世纪 80 年代，数字移动通信系统诞生，称为**第二代移动通信网络**(2nd Generation MCN，2G)。从 2G 开始，数据业务逐步拥有了与语音业务同等重要的地位。分组交换、文本短消息、多媒体信息、宽带接入 Internet 等成为移动通信网络所具备的主要功能。随着频带复用、信道编码等无线通信技术不断进步，频率资源利用率和数据通信速率获得不断提高，在

经历了 2.5G、2.75G 的技术升级后，**第三代移动通信网络**(3rd Generation MCN，3G)应运而生。按照技术发展的节奏，4G、5G 时代的到来应当也为时不远了。数字移动通信系统的技术演进如图 7.3 所示。

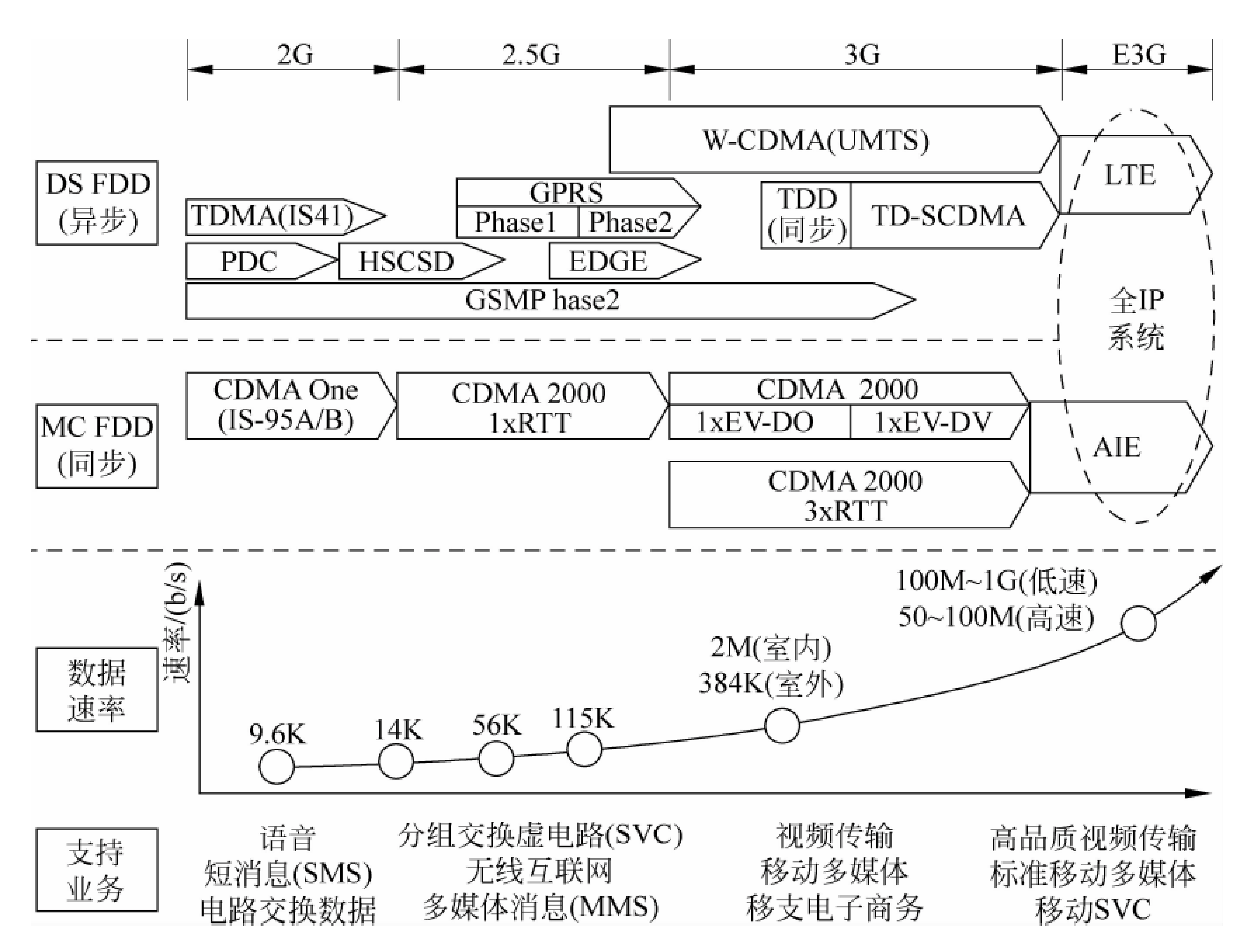


图 7.3　数字移动通信网络技术演进示意

移动通信系统 NSS 内部或 NSS 与外部系统和网络之间的接口采用 No.7 信令实现，另外还采用 MAP、ISUP、TUP 等不同应用层类型的信令协议。

7.2　移动通信网络关键技术

7.2.1　号码管理

如图 7.4 所示，移动通信系统需对用户设备进行编号，以便在移动性管理和呼叫时迅速、准确、安全地识别目标。编号由不同设备记录并用于不同协议。

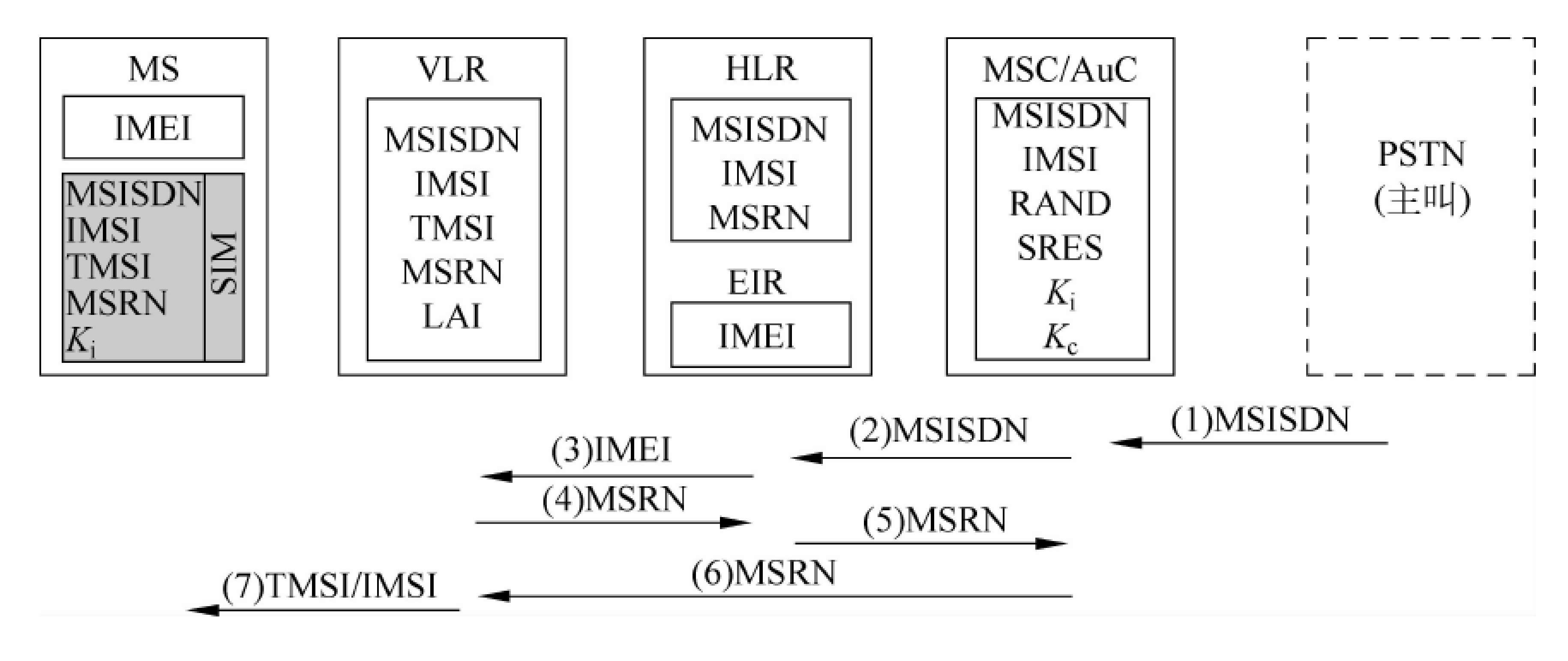


图 7.4　移动通信系统号码管理和使用示例

1. IMSI

国际移动用户识别码(International Mobile Subscriber Identity,IMSI)永久性唯一地属于一个移动用户,包括漫游区域在内的所有位置都是有效的。IMSI 储存在 SIM 卡及 HLR 等设备中,用于位置更新、呼叫建立等信令协议,出于安全保密的考虑,一般不在无线接口上传送。IMSI 采用 ITU-T E.212 编码规则,共由 15 位数字组成,包括如下内容。

(1) **移动国家号码**(Mobile Country Code,MCC),3 位,中国是 460。

(2) **移动通信网号码**(Mobile Network Code,MNC),2~3 位,由每个国家分配。例如中国移动是 00 或 02,中国联通是 01。

(3) **移动用户识别号**(Mobile Subscriber Identification Number,MSIN),最多 10 位,由运营商分配,唯一识别该运营商网络中的移动设备或用户。

MNC 和 MSIN 共同构成一个国家内唯一识别移动用户的**国内移动用户识别号**(the National Mobile Subscriber Identity,NMSI)。

2. TMSI

临时移动用户识别码(Temporary Mobile Subscriber Identity,TMSI)由 VLR 分配给位置登记或位置更新后的来访移动用户,与 IMSI 唯一对应,但 TMSI 仅在该 VLR 所管理的区域使用。TMSI 为 4B 的 BCD 码,即 8 个十六进制数。TMSI 在呼叫建立和位置更新时在空中接口中使用,起到安全替代 IMSI 的作用。

3. MSISDN

移动用户国际电话号码(Mobile Subscriber International ISDN/PSTN Number,MSISDN)是用于移动通信网络呼叫拨号的号码(即手机号码)。MSISDN 采用 ITU-T E.164 编码规则,存储在 HLR 和 VLR 中,在移动接入网协议接口中传送。MSISDN 由最多 16 位数字组成。

(1) 国家码(Country Code,CC),中国为 86。

(2) 国内接入码(National Destination Code,NDC),3 位数字。例如,中国移动有 139、138、158 等,中国电信有 189 等。

(3) 用户号码(Subscriber Number,SN),由运营商负责分配,组成方式为 $H_0H_1H_2H_3$ABCD,其中,$H_0H_1H_2H_3$ 为 HLR 识别号,$H_0H_1H_2$ 由全网统一分配,H_3 为省内分配,ABCD 为一个 HLR 中移动用户的号码。

4. MSRN

移动站漫游号码(Mobile Station Roaming Number,MSRN)用于移动终端漫游时进行路由选择,将呼叫转移到移动终端当前所登记的 MSC,并由 VLR 临时分配一个 MSRN,呼叫接续完成后即释放。

5. 其他号码

其他号码包括本地移动用户识别码 LMSI、切换号码 HON、位置区识别码 LAI、漫游区域识别码 RSZI、基站识别码 BSIC、国际设备识别码 IMEI(唯一标识手机)等。

7.2.2 用户鉴权

用户鉴权用于防止未授权的非法终端接入移动通信网络。鉴权的依据是移动终端中的 SIM 卡存储的 IMSI、TMSI、K_i、K_c 等参数和加密算法 A_3、A_8,这些数据是在 SIM 卡合法登记开通时由网络运营商加密写入并在**鉴权中心**(AuC)中记录。

用户鉴权密钥 K_i 用于 A_8 加密算法，A_3 为鉴权算法，且 A_3 是保密的。用户鉴权过程采用 AuC 生成的鉴权三参数组来实现：

{随机数 RAND，符号响应 SRES，加密密钥 K_c}

RAND 为 128b 随机数；SRES 为 32b，由 RAND 和 K_i 经 A_3 鉴权算法获得；K_c 则由 RAND 和 K_i 经 A_8 加密算法获得。鉴权三参数组产生流程如图 7.5 所示。

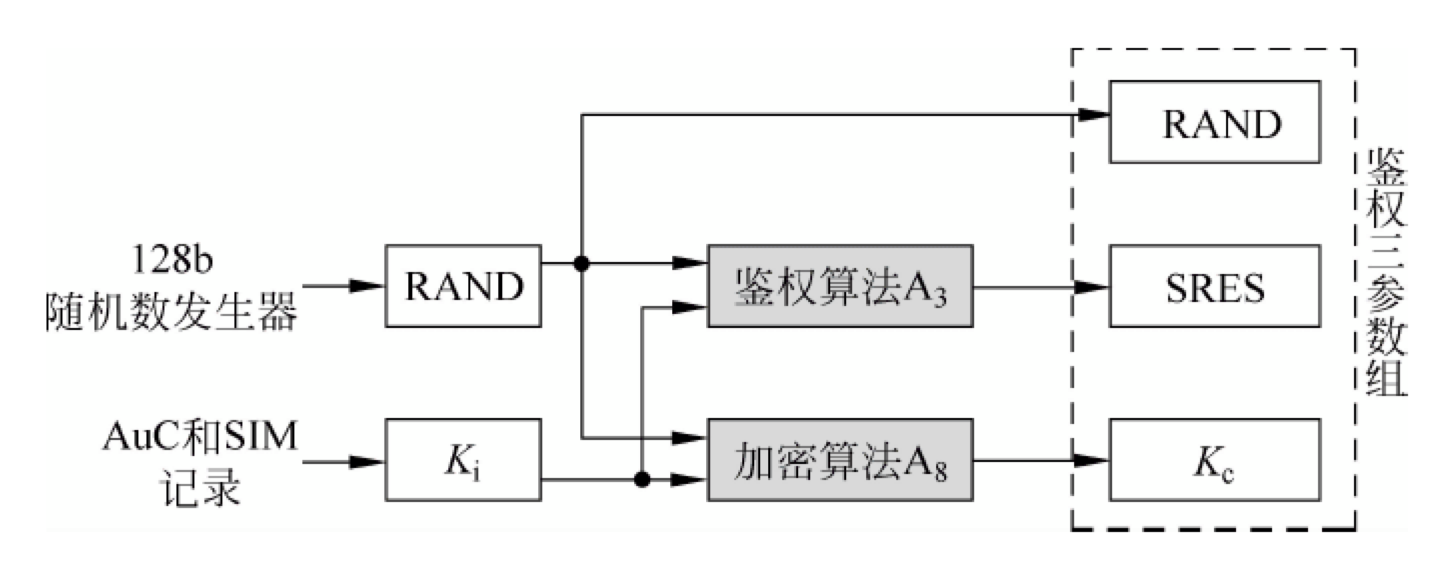


图 7.5 鉴权三参数组产生流程

用户终端每次在进行接入登记、呼叫建立、位置更新等操作前均需要先完成鉴权。鉴权流程如下。

(1) 当移动终端请求接入时，MSC/VLR 通过控制信道将三参数组中的 RAND 参数传送给终端。

(2) 终端收到 RAND 后，用此 RAND 和 SIM 卡中存储的 K_i 经同样的 A_3 算法计算获得一个符号响应 $SRES_{MS}$，并将其传回 MSC/VLR。

(3) MSC/VLR 将收到的 $SRES_{MS}$ 与记录的三参数组中的 $SRES_{AuC}$ 进行比较，如果相同就允许用户接入，否则拒绝接入。

7.2.3 用户漫游

用户漫游(Subscriber Roaming)是移动用户及其终端在不同地理位置上转移的过程，就移动通信网络而言，就是移动终端位置管理的问题。所以，漫游管理等价于定位(位置)管理。

用户漫游包括多种情况：在新位置开机，在开机状态移动到新的位置等。漫游还分为同一 MSC 内不同 BSC 的位置更新以及不同 MSC 间的越局位置更新。

例如，移动终端从原来的 MSC1 漫游到当前的 MSC2，则应当在新的位置上进行更新操作，基本流程如图 7.6 所示。

7.2.4 无缝切换

切换(handover)是指将一个处于呼叫建立状态或通话状态的移动终端转换到新的信道上，并保持已经建立的链路不被中断的过程。由于移动用户感觉不到切换操作，因此常称为**无缝切换**(seamless handover)。

切换一般由移动中断监测并判决，交由交换中心控制完成。在切换过程中，终端和基站均参与其中，切换与否则由基站决定。

移动终端在通话过程中，不断向所在小区的基站报告本小区和相邻小区的无线环境参数，同时 BTS 也在不停地测量上行信号质量、强度及时间提前量，BTS 将测量报告送往基站

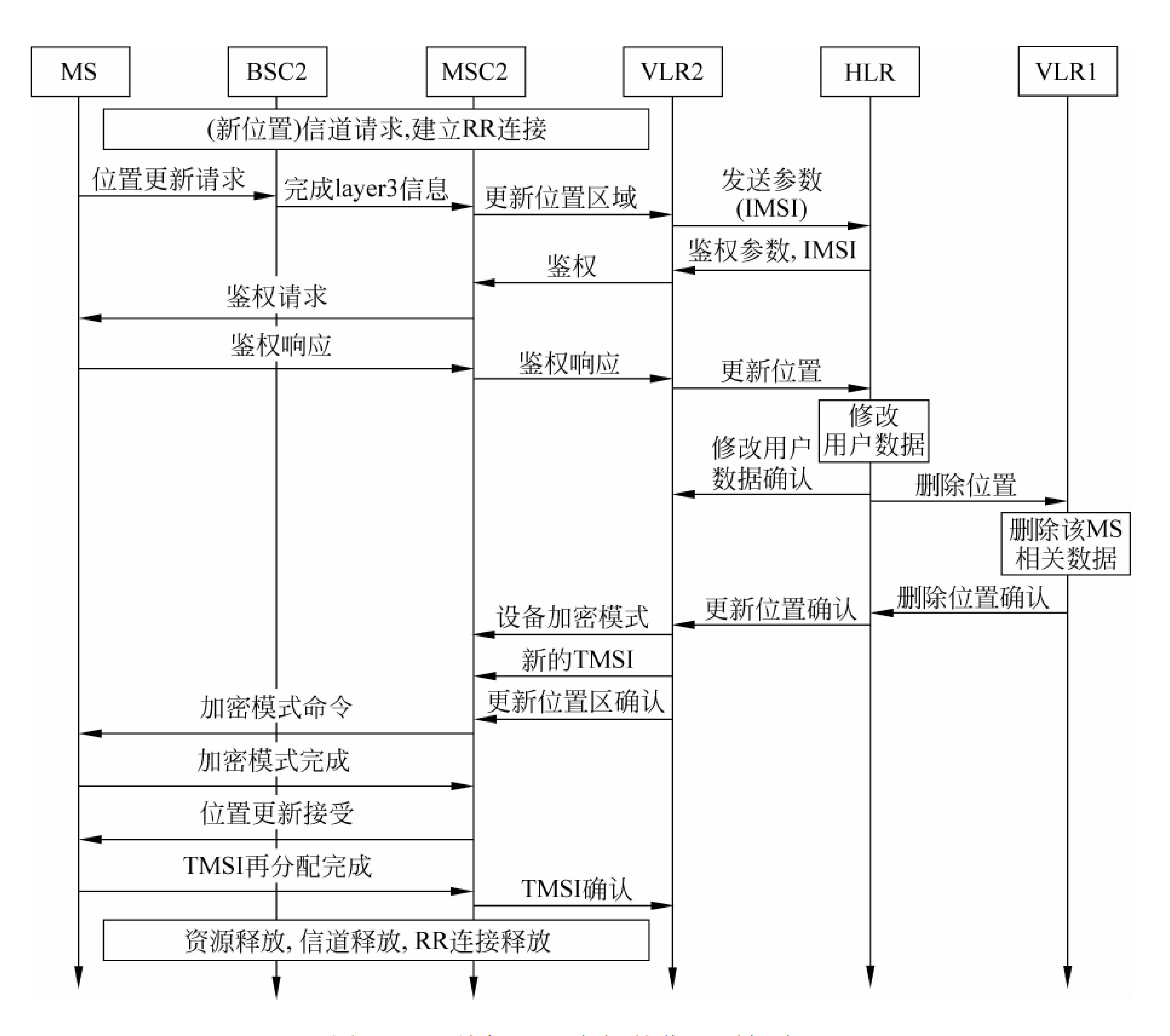


图 7.6　不同 MSC 之间的位置更新流程

控制系统(BSC),BSC 根据这些信息对各个小区状况进行排序,最终决定是否需要切换,以及切换到哪个 BTS。

切换可分为 BSS 内部切换、同 MSC 内不同 BSS 间切换、MSC 间切换。如图 7.7 所示为不同 MSC 间切换的信令流程。

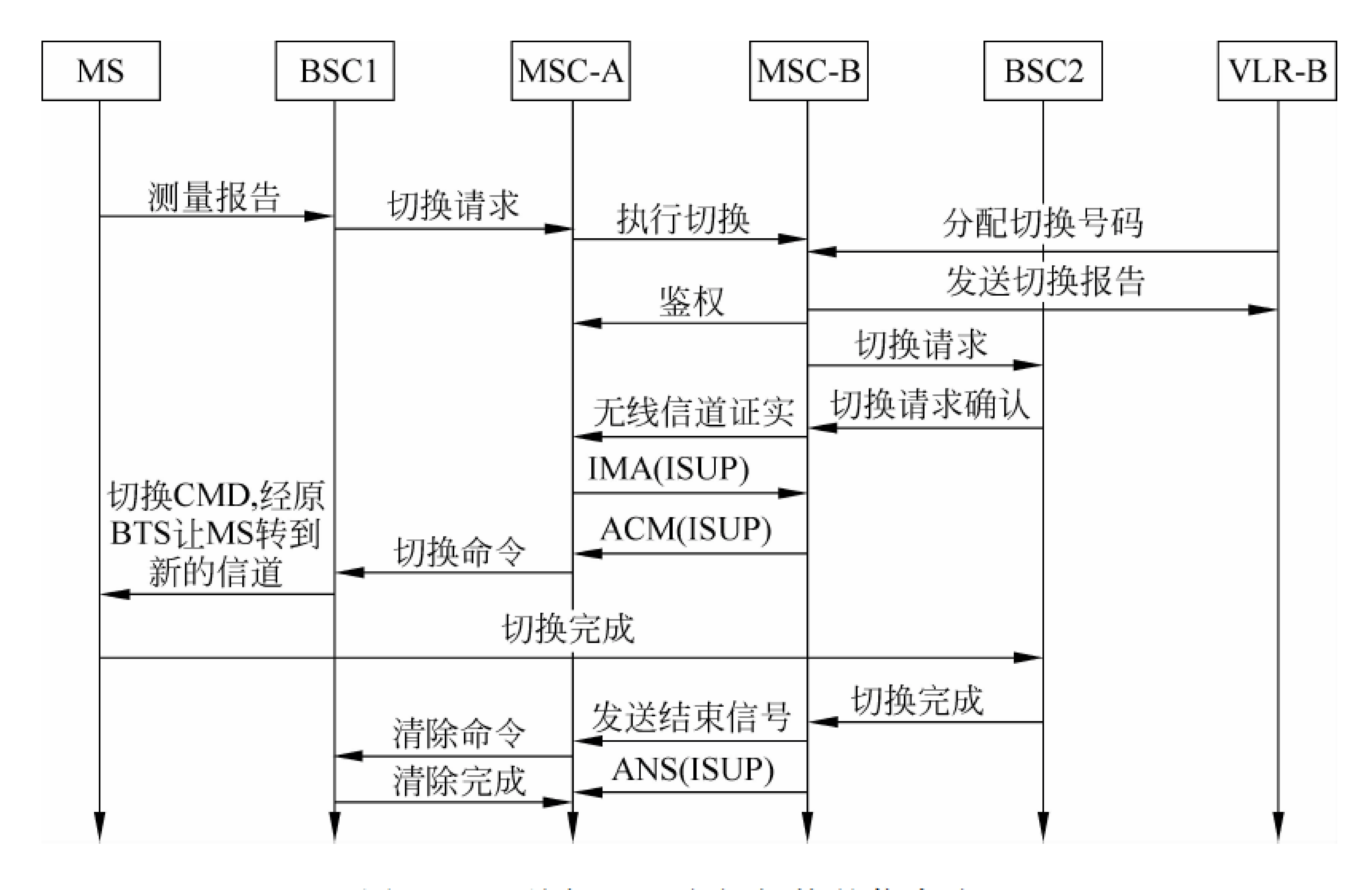


图 7.7　不同 MSC 之间切换的信令流程

7.3 2G 网络

从 2G 网络开始,移动通信技术和网络进入数字化时代,与第一代模拟系统相比,有了质的飞跃,为进一步的发展奠定了良好的基础。由于 2G 网络技术非常成熟、网络覆盖面大、用户保有量多,因此,即使在如今的 3G 时代,仍然有许多网络、许多用户"停留"在 2G 上。究其原因,用户不愿意立即更换手机、3G 网络吸引力不够、3G 网络建设尚未完全到位是主要因素。

2G 网络采用 TDMA 或 CDMA 多址接入技术,其技术主要分为两大阵营:欧洲的 GSM 和北美的 CDMA。2G 网络技术指标和地区分布情况如表 7.1 所示。

表 7.1　2G 技术指标和网络分布

移动通信网络技术	IS-95	GSM	IS-54	PDC
分布地区	美国、中国	欧洲、中国	美国	日本
多址访问方式	CDMA	TDMA/FDD	TDMA/FDD	TDMA/FDD
调制方式	QPSK/OQPSK	GMSK	π/4 DQPSK	π/4 DQPSK
前向信道(MHz)	869～894	935～960	869～894	810～826
反向信道(MHz)	824～849	890～915	824～849	940～956
信道间隔(kHz)	1250	200	30	25
数据/码片速率(Kb/s)	1.2288(Mcps)	270.833	48.6	42
语音编码速率(Kb/s)	1.2～9.6	13.4	7.95	6.7

GSM(Global Standard for Mobile Communication)采用 900MHz 频段,频分双工(FDD)方式,分为下行(前向)链路和上行(反向)链路,每条链路两边各留一段频段,防止相互干扰。如图 7.8 所示,两条链路共分为 124 对载频,每路载频用 TDMA 方式划分出 8 个时隙,这样,GSM 共可获得 992 个双工传输信道。

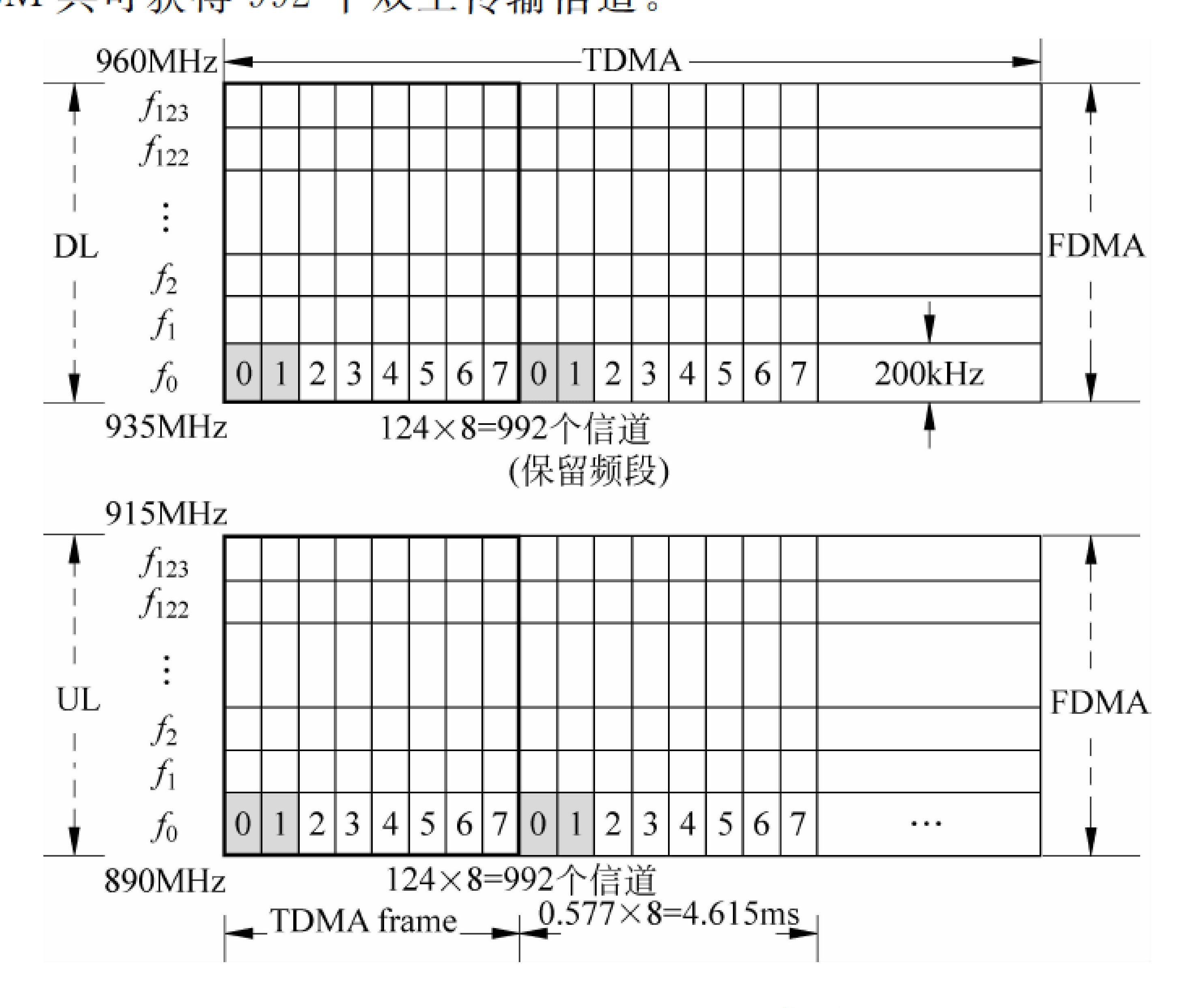


图 7.8　GSM 上行和下行信道

主载频 f_0 的时隙 0 和 1 用于传送信令，为控制信道；其余信道用于传送语音或数据，为业务信道。数据传输的最高速率为 9.6Kb/s。常用的**短消息服务**(Short Message Service，SMS)是通过控制信道传送的。

通用分组无线业务(General Packet Radio Service，GPRS)是 GSM Phase2.1 规范实现的内容，引入了分组交换功能，能够提供比 GSM 网络业务信道更高的数据速率。GPRS 与 CDMA1x 一起也被称为 2.5G 网络。

GPRS 采用与 GSM 系统相同的频段、频带宽度、突发结构、无线调制技术、跳频规则和相同的 TDMA 帧结构，但在信道分配、接口方式、数据传输等方面体现了分组业务的特点。如图 7.9 所示，GSM 网络可以通过增加 PCU、SGSN、GGSN 等 3 个网元设备，平滑升级到 GPRS 网络。

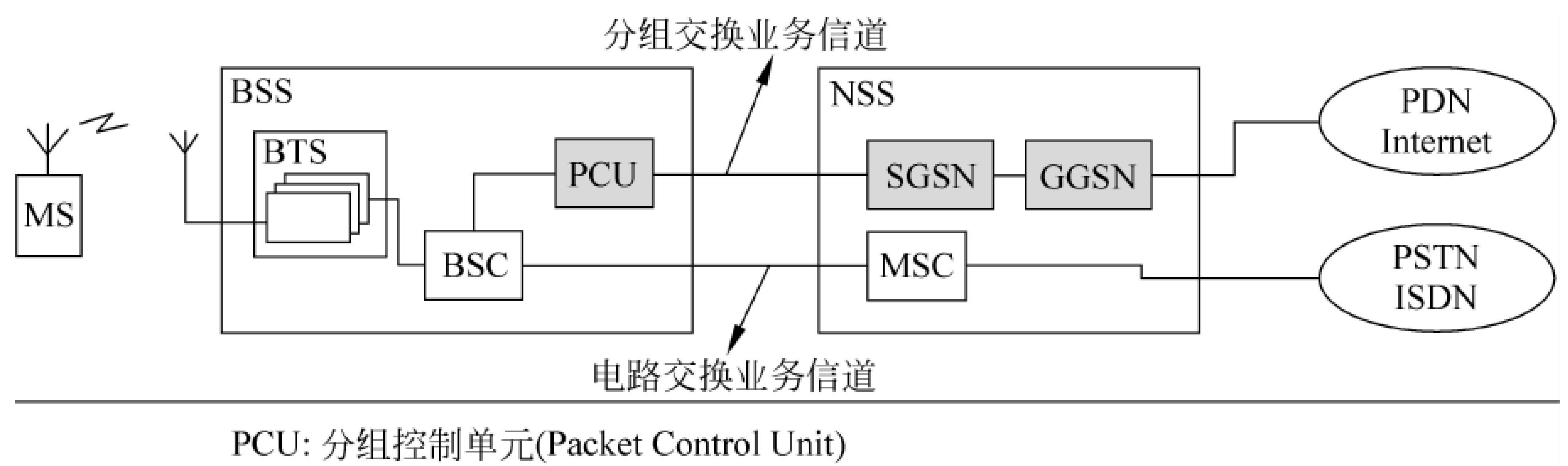


图 7.9　GPRS 系统示意

GPRS 可以提供高达 171.2Kb/s 的无线接入峰值速率(8 个时隙)。

增强型数据速率 GSM 业务(Enhanced Data rates for the GSM Evolution，EDGE)经常被称为 2.75G 移动通信网络，用于提供 GSM/GPRS 系统上的高速数据服务。

EDGE 兼容 GSM/GPRS 网络，在保持 200kHz 载频带宽不变的前提下，通过改变调制、编码方式提高数据传输速率，同时对核心网影响很小。

EDGE 技术包括：基于电路交换的 GSM 网络的增强电路交换数据业务 ECSD、基于分组交换 GPRS 网络的增强通用分组无线业务 EGPRS、基于 D-AMPS 网络的增强技术 EDGE 压缩技术。

由于 ECSD 和 EDGE 压缩技术存在一定的技术局限性，未得到应用，故 EDGE 通常是指 EGPRS 技术。对一个多时隙的 EGPRS 终端，同时使用 8 个时隙，可达到无线接入的最大数据速率 475Kb/s。

7.4　3G 网络

3G 移动通信系统被 ITU-T 称为 IMT-2000，北美称为 CDMA-2000，在欧洲则称为 UMTS(Universal Mobile Telecommunication System)。3G 规范由第三代移动通信合作伙伴 3GPP 和 3GPP2 负责制定。

3G 网络基于 2G 的 GSM/CDMA 核心网结构，但是采用新的空中接口协议，工作于 2GHz 频段。

3G 网络有三类共五种技术标准。

(1) WCDMA。宽带码分多址(Wideband CDMA,WCDMA)支持两种基本的双工方式:频分双工(FDD)和时分双工(TDD),采用宽带直扩码分多址(DS-CDMA)技术。

快速移动环境中,WCDMA 最高速率为 144Kb/s,步行情况下达 384Kb/s,静止时达到最高 2Mb/s。而采用 HSDPA 技术,下行速率可达 10.8Mb/s,HSUPA 的上行速率达到 1.4～5.8Mb/s。

WCDMA 中的**软切换**(soft handover)技术富有特点,与硬切换相比,由于软切换可同时与多个小区保持通信、接收端利用宏分集技术降低接收信号衰落的概率,有助于降低掉话率。

(2) CDMA2000。CDMA2000 的核心网和无线网技术的改进是分阶段、各自独立进行的。CDMA2000 一个载波的带宽为 1.25MHz,如果分别使用每个载波,被称为 1x 系统,如果 3 个载波捆绑使用,则为 3x 系统。CDMA2000 的演进方向是 1x/EV-DO 和 1x/EV-DV。

(3) TD-SCDMA。时分同步码分多址(Time Division Synchronous CDMA,TD-SCDMA)是中国主导研发和提出的 3G 国际标准,具有重要的国家战略意义。

TD-SCDMA 核心网与 WCDMA 基本相同,不同处在于无线接入网络部分。TD-SCDMA 无线传输方案是 FDMA、TDMA 和 CDMA 三种多址技术的结合运用,具有较显著的特色(如图 7.10 所示)。

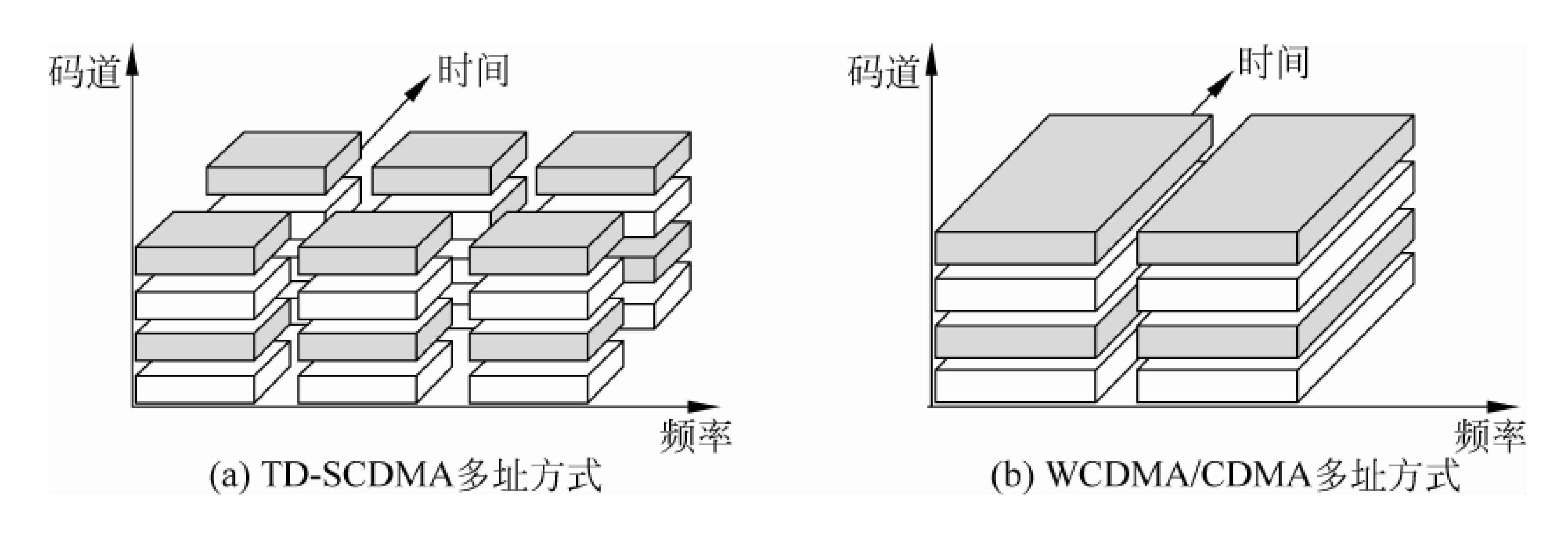


图 7.10　TD-SCDMA 多址方式技术特色

TD-SCDMA 采用 TDD 双工方式,可动态分配信道,通过周期性地转换传输方向,允许在同一个载波上交替地进行上下行链路传输(就像城市交通中的可变车道),理论上仅需单载波就可提供 2Mb/s 数据速率。

TD-SCDMA 运用了**智能天线**(smart antenna)技术,基于自适应天线阵原理,利用天线阵的波束赋形,产生多个独立的波束,并自动调制波束方向来跟踪每一个用户,可减小干扰、增加系统容量、降低发射信号功率。智能天线还属于一种空分多址技术。此外,TD-SCDMA 还应用了上行同步、联合检测、接力切换等先进技术。

从图 7.11 可以看到,3G 不是移动通信网络发展的终点。以 3G 为新的起点,移动通信技术已经演化出 E3G/S3G、B3G(Beyond3G,超 3G),并向 4G 迈进。

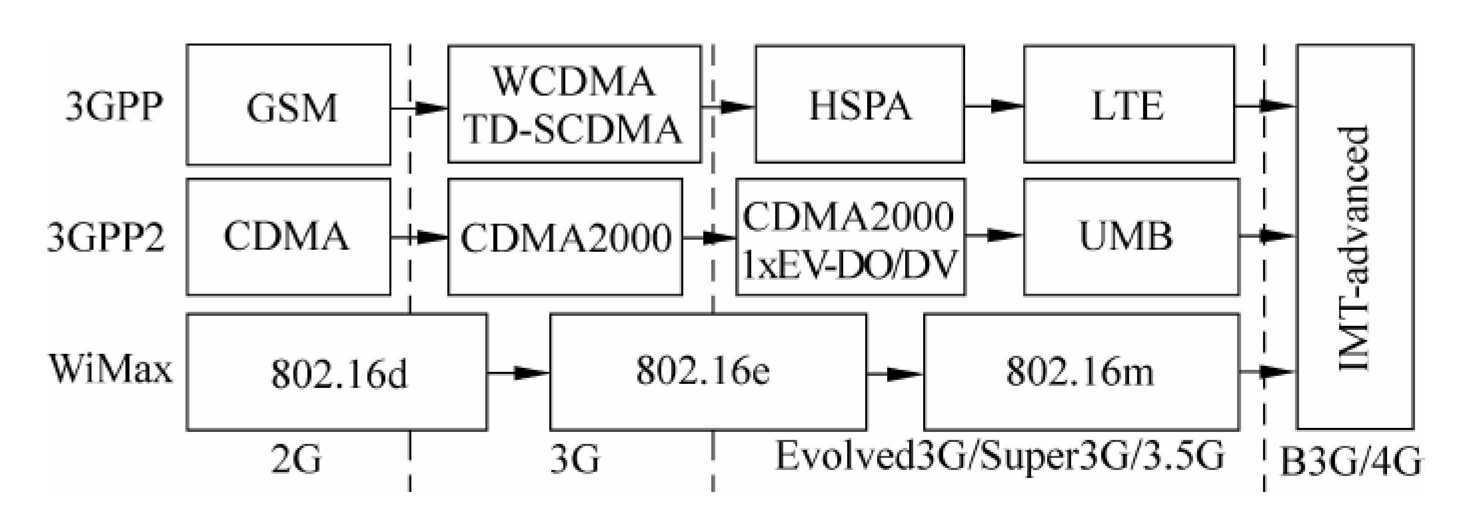


图 7.11　3G 演进路线

7.5　WAP

无线应用协议(Wireless Application Protocol,WAP)是移动通信系统上的信息访问技术框架,为移动通信用户(尤其是手机)提供类似 Internet 信息浏览的服务。

WAP 有 WAP 1.x、WAP 2.0 和开放移动联盟(OMA)3 个发展阶段。WAP 1.x 的应用目标是针对非智能手机的系统结构要求,而随着移动通信网络发展到 2.5G,智能手机也逐渐普及,WAP 开始与 HTML 进行融合,以适应更加具有开放性的业务需要。

WAP 体系主要由 WAP 代理和 WAP 网关构成(如图 7.12 所示)。

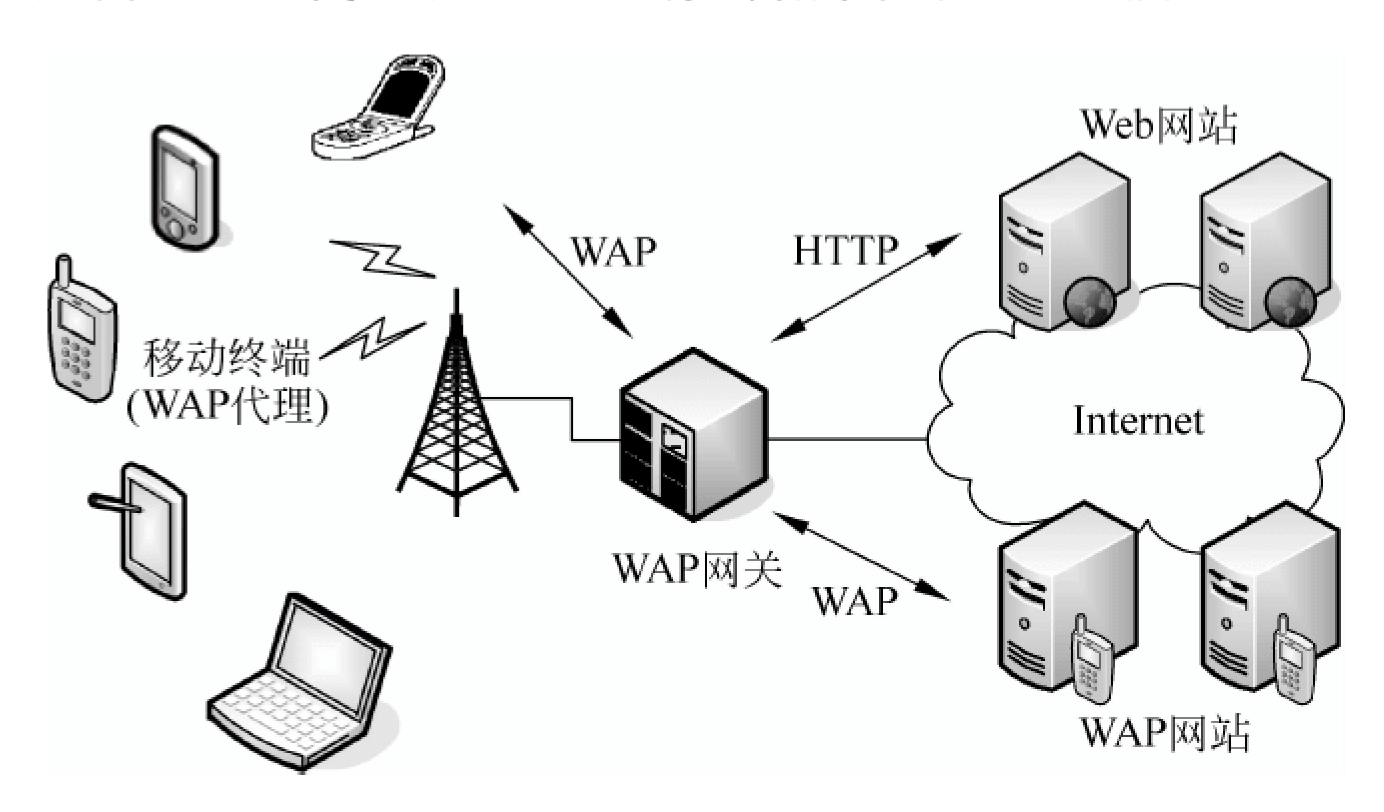


图 7.12　WAP 体系结构示意

WAP 代理即安装在手机上的微型浏览器,可以使用 WAP 体系定义的无线标记语言(Wireless Markup Language,WML)和脚本语言(WML Script,WMLS)来访问 WAP 网站。WML 是一种与 HTML 非常相似的超文本描述语言,采用标签定义的结构。

WAP 网关主要实现协议适配和信息编解码功能。通过 WAP 网关,可以将 WML 转换为适合在无线环境中传输的压缩二进制码,减少带宽占用。WAP 网关可支持从 WAP 到 HTTP、WML 到 HTML 的转换,使 WAP 用户可以访问 Internet 上的大量网站资源。

WAP 提供两种信息访问模式:Pull(**拉**)模式和 Push(**推**)模式。

Pull 模式即常用的用户主动发起浏览信息请求,服务器进行响应的方式。Push 模式则是 WAP 技术所独有的,业务运行在 WAP 协议栈会话层协议之上,由网络主动发起向用户发送信息,可将信息推送到手机上(与 SMS 的推送模式相似)。

俗称彩信的**多媒体消息业务**(Multimedia Message Service,MMS)是 SMS 和增强型 SMS(EMS)的进一步发展,也是 WAP 体系的承载业务之一。此外,WAP 还可以支持个人信息助理(Personal Information Manager,PIM)个性化服务和流媒体点播、互动游戏等丰富的应用类型。

在网络应用中,是否可将信息主动推送到用户终端上,不仅关系到信息能否被用户及时掌握,而且关系到商业模式及其价值。例如,许多软件厂商想方设法占领用户计算机桌面,使得广告等内容可以强制性地展现出来,获取用户的注意和点击,谓之眼球经济。然而,粗暴的、非自愿性的推送是不恰当的,而优雅的、两厢情愿的推送则可以增强应用的效果,如:PushMail 可提醒用户立即处理新到达的邮件,手机短信、彩信亦然;客户端软件或插件可自动抓取、更新用户所需要的新闻等。

CHAPTER 8

第8章 多媒体信息编码

8.1 信息编码原理

多媒体(Multimedia)是指数字化的人体可感知信息,包括文本(text)、图形(graphics)、图像(image)、动画(animation)、音频(audio)和视频(video)等,面向视觉和听觉感官。其他诸如触觉、嗅觉、味觉等感知的数字化技术尚在研究中。此外,媒体(media)的概念中还包括表示媒体(信息编码)、显示媒体(输入输出)、存储媒体和传输媒体。

多媒体信息是网络的最终表现形式,是所有网络服务的目标。多媒体信息具有数据量大、类型丰富、集成度高的特点,对承载多媒体信息传输的网络提出了很高的要求,网络应当具备宽带传输和 QoS 支持能力,能够满足多媒体信息系统的实时性、交互性需要。

多媒体网络的关键技术由两个方面组成:多媒体信息的编码技术,可以将模拟媒体数字化,并且在保证一定质量的前提下尽可能减小数据量,以有利于网络的高效传输;多媒体网络的设计技术,可以针对多媒体传输的技术特性和应用需求,发挥网络优势并且发掘技术潜力,达到多媒体信息的处理、传输和表现的最佳效果。

多媒体信息**编码**(coding)是多媒体信息在网络系统中从生成到表现的整个过程中的重要环节。如图 8.1 所示,信息编码关系到数字化多媒体信息的获取和压缩,是其他后续过程的基础。

多媒体信息编码由两方面技术组成:一是对采集到的模拟和数字信息采用数字编码方式来表达;二是对生成的数字编码进行压缩,以减小数据量,有利于存储、传输和处理。

模拟信号的数字化可采用 PCM 等 A/D 转换技术,如声音、环境数据。数字信息主要是指文字(汉字、英文字母、数字、符号等)、图形和图像传感器(如 CCD)采集的点阵数据(色彩、亮度)。

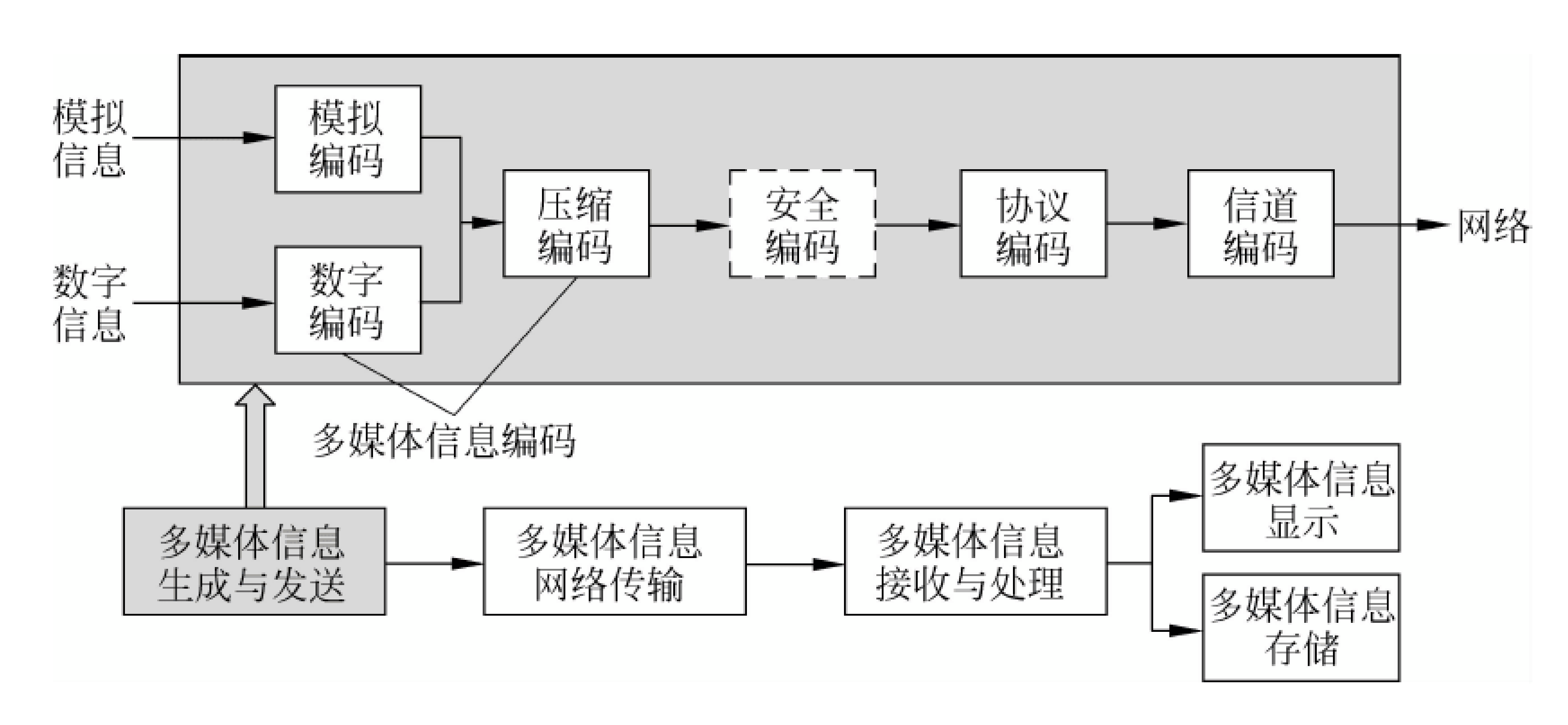


图 8.1　信息编码环节示意

信息编码的重点是数字**压缩**(compressing)技术。数字压缩的依据主要是信息的数字化表达存在一定的冗余性，包括空间冗余、时间冗余、编码冗余、结构冗余、知识冗余和感官冗余。

(1) 空间冗余。空间冗余一般存在于图像数据中。例如，在一幅图像中，一些局部区域的颜色、灰度值比较接近，形成属性相似的集合，就具有空间(或空域)上的强相关性。如一个 16×16 的图像块，共 256 个数据点，十进制数值在 122～129 之间，每个数值需要用 8b 表示，总共需要占用 2048b，而如果表示为以 122 为基数的数值，每个点位只需用 3b 表示，共 768b，在保持原有信息量完全不变的前提下，数据量减少了 62.5%。

利用空间冗余的编码技术主要有变化编码、帧内预测编码等。

(2) 时间冗余。时间冗余是动态视频和音频数据中存在的冗余。相邻的两幅图像一般有较大的相关性，音频也是一个逐渐变化的过程。时间轴上前后两个相关的数据集合(如相同位置的图像块)，可能是完全相同的，或只是产生了非常小的变化。例如，视频中往往只有一个物体在运动，而其他背景图像是不变的，那么可以参照和引用前一时间的数据，而不必进行重复，从而在总体上减少了数据量，被称为时间压缩。

帧间预测编码方法(运动估计和补偿)是利用时间冗余的主要编码技术。

(3) 编码冗余。编码冗余又称为信息熵冗余。如果对多媒体的所有信息元素都用相同长度的符号来表示，在编码上必然存在冗余性。理想情况是按照每个元素信息熵的大小为其分配相应的二进制比特数。例如根据每个数值出现的概率来分配，出现的概率越高，其编码长度越小，那么整体数据量就可大大降低。

以信息量观点来看，事件概率越大，其信息量越小，其占用的数据量也应该越小(编码长度短)。设图像点阵数据构成离散符号集$\{x_i\}$($i=1,\cdots,n$)，独立概率分别为 $p(x_i)$，且有 $\sum p(x_i)=1$，则信息熵 $H(x)$为(以 b 为单位)：

$$H(x)=-\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

在空间冗余的例子中，符号集为$\{122,\cdots,129\}$，假设各符号出现的概率均等，即 $p(x_i)=1/8$，熵达到最大值$\log_2 8=3$(b/symbol)，与空间冗余分析所得结果完全相同。当然，各符号出现的概率在实际应用中并不相同，但采用编码冗余的思想，就可以使数据量尽可能逼近信息熵。

采用编码冗余的技术主要有霍夫曼编码、算术编码、游程编码、二进制算法编码和基于上下文的熵编码等。

(4) 结构冗余。结构冗余是指图像本身构造上的冗余。可采用轮廓编码、区域分割等方法,一般为有损压缩技术。

(5) 知识冗余。知识冗余指人们已知的某些图像所具有的先验知识、背景知识。如果某些规律可以从这些知识中获取,则相关数据就成为冗余信息。

(6) 感官冗余。感官冗余又称心理感觉冗余,指人类听觉和视觉感官无法察觉的信息。例如听觉系统对2k～5kHz的声音最敏感,那么在这个范围之外的频率信息对人类来说就不太重要,变成冗余信息;人类的视觉系统对亮度比色度更敏感,对变换后的低频信号比高频信号更敏感,能够分辨的图像灰度等级约为2^6个等,利用这些特点可以采用有针对性的量化方法,有效减少数据量。由于基于感官冗余的编码舍去了一些信息,因此是一类有损压缩技术。主要方法有差分脉冲调制编码、变换编码、标量量化与矢量量化编码、模型编码、多分辨率编码等。

如图8.2所示,编码系统由发送方的编码器(coder)和接收方的解码器(decoder)组成,编码器包括分析模块、量化模块、二进制编码模块,解码器则包括对应的二进制解码模块、反量化模块和综合模块。分析模块通常采用预测编码和变换编码。量化模块有两种量化类型:均匀量化和非均匀量化。二进制编码模块一般采用熵编码技术。

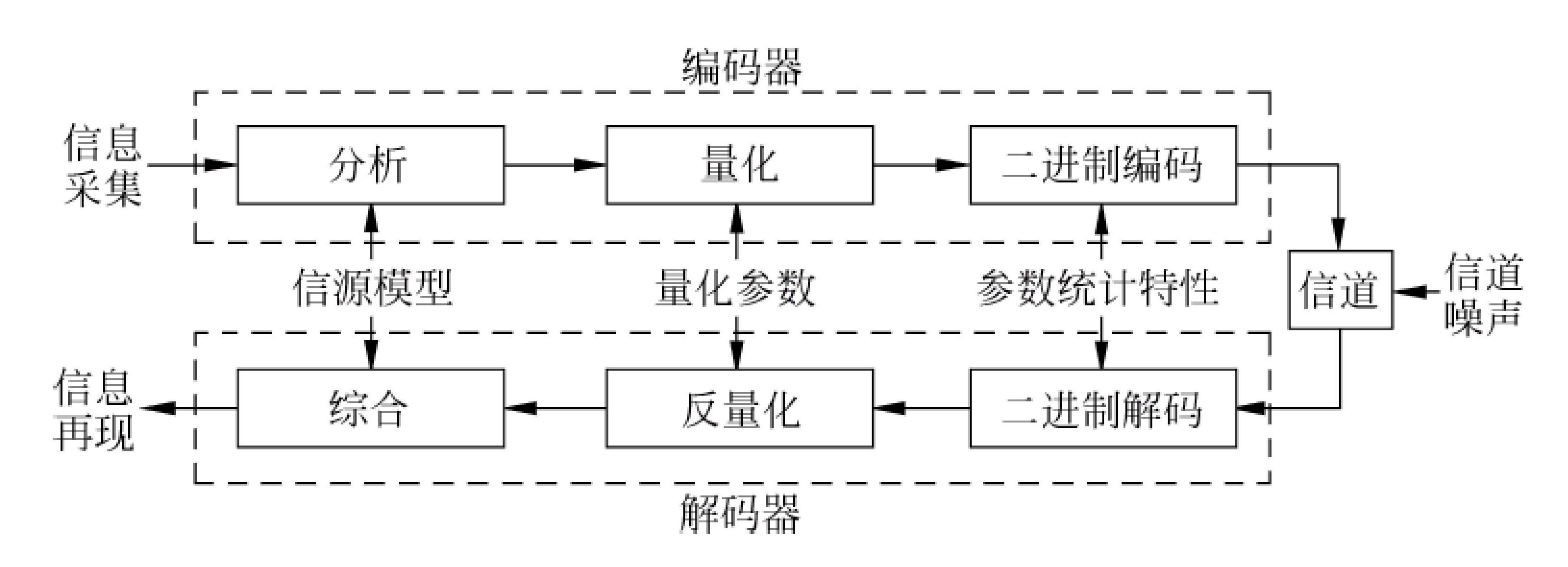


图8.2 编码系统组成结构

8.2 信息编码算法

8.2.1 霍夫曼编码

霍夫曼编码(Huffman Coding)是消除编码冗余的最常用方法,也常见于其他信息系统编码方案中。霍夫曼编码的理论依据是可变字长理论,对出现概率较大的符号用较短的字长,反之用较长的字长。霍夫曼码被称为最优码,因为对于给定的符号集和概率模型,找不到任何其他比霍夫曼码更短的平均字长。

对于n个不同概率的信息符号,各符号的初始编码均为空,每次编码长度增加1b,往高位延长,霍夫曼编码算法各步骤如下(如图8.3所示)。

(1) 将所有信息符号按概率大小排序。

(2) 两个最小概率对应的符号分别延长编码0和1(对已合并的符号均延长相同比特),并将概率相加、符号合并。

(3) 若最后只剩下两个符号，分别延长编码 0 和 1，算法结束；否则返回步骤(1)继续。

由于两个最小概率的符号可任意分配 0 和 1，概率相同的两个符号也可任意排序，因此对同一个信源的霍夫曼编码并不唯一，但得到的平均码长相同。一旦获得霍夫曼编码后，编码和解码(见示例程序)可通过简单的查表方法实现。

```
symbol sym - tab[m] = {S0,S1, … };
binary huff - tab[m, n] = {{0},{1,1}, … };
symbol decoder() {
int i,j = 0;
  for i = 0 to m - 2 do
    if (next - bit() == huff - tab[i, j])
    then return(sym - tab[i]);
    else j++;
  return(sym - tab[m - 1]);
};
```

霍夫曼编码是无损压缩方法，编码没有歧义性，只有唯一的解码方法；每个符号的解码并不依赖其他符号，是一种即时码。

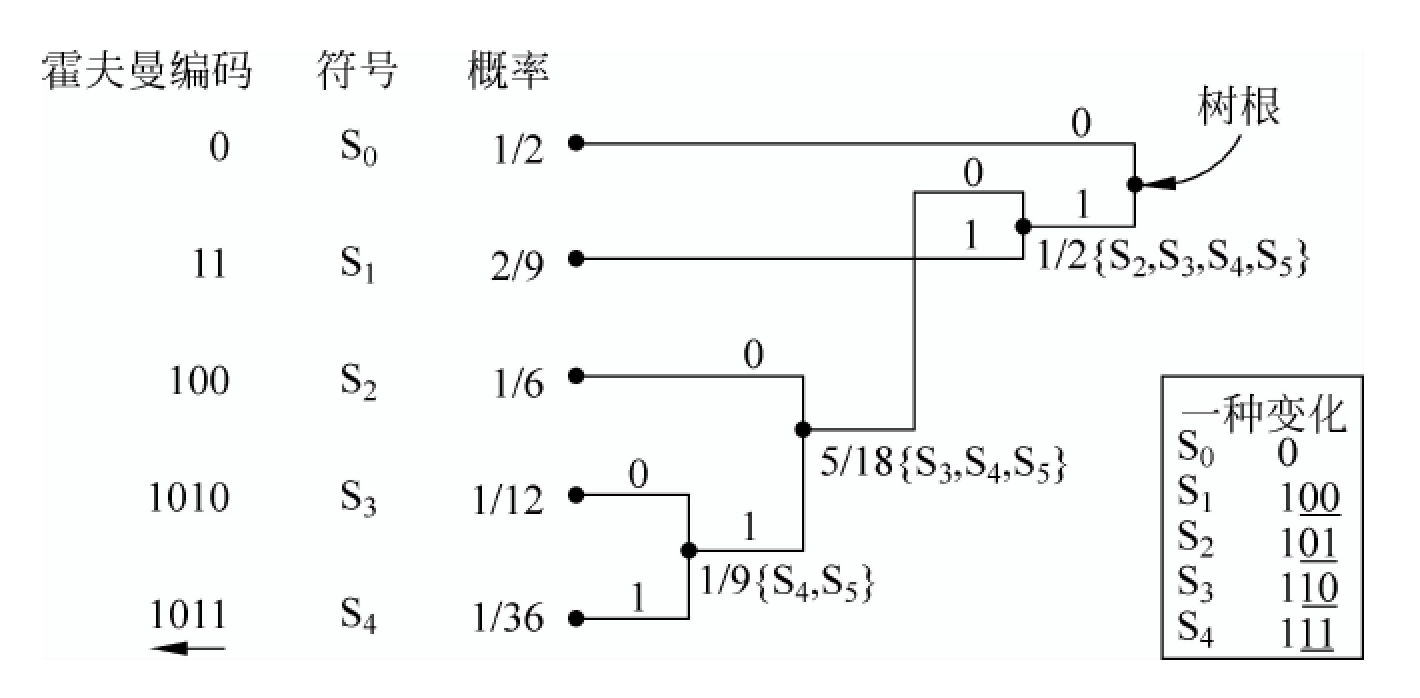


图 8.3　霍夫曼编码示例

8.2.2　游程编码

游程编码(Run Length Coding)又称为游程长度编码，是一种利用空间冗余压缩多媒体数据的方法，属于统计编码类。

游程编码比较适合二值图像，每个像素用 0 和 1 表示。这样，原始图像数据序列就必然由交替出现的连续 0 和连续 1 组成，例如，1111001111111011。游程 $L(0)$就是连续 0 的个数，$L(1)$就是连续 1 的个数，均为大于等于 1 的整数，那么，游程编码就是 $L(0)$和 $L(1)$的交叉排列，例如，42712。显然，这一变换是可逆的，而且具有唯一性。

游程编码是图像的无损压缩方法，还可以用霍夫曼编码进一步压缩(长度值用霍夫曼编码表示)。但游程编码对于连续性不足的图像可能产生编码后数据量增加的情况，应与其他图像编码方法结合使用。

8.2.3　算术编码

算术编码(Arithmetic Coding)是用于视频的压缩方法，采用实数表达而非通常用的整数，是一种能够逼近熵极限的最优算法。

算术编码的基本思想是从整个信源的符号序列出发,采用递推形式进行连续编码,整个信源符号序列使用一个算术码字表示,而码字为介于 0 和 1 之间的实数。码字所代表的实数区间,随着信源符号序列中符号数目的增加,区间长度不断减小。

设符号集$\{S_0, S_1, \cdots, S_i, \cdots\}$对应的概率为$\{P_0, P_1, \cdots, P_i, \cdots\}$,应有$\sum P_i=1$。各符号对应的积累概率$P_{ci}$是指$\sum P_j, j=0, \cdots, i-1$,其中$P_{c0}=0$。由此,得到各符号$S_i$的初始编码区间为$[P_{ci}, P_{ci}+P_i)$。

对输入符号序列,依次编码各个符号(反复执行编码过程),直到遍历所有符号。算术编码过程可描述为如下的一个递归过程。

(1) 初始状态

编码点变量:$P_c=0.0$

编码区间变量:$P=1.0$

(2) 编码过程

新编码点:$P_c=P_c+P\times P_{ci}$

新编码区间:$P=P\times P_i$

最终输出P_c即为符号序列的算术编码结果。

【例 8.1】 有信源符号为$\{S_0, S_1, S_2, S_3\}$,符号概率分别为$\{0.1, 0.4, 0.2, 0.3\}$,据此,可把区间[0,1)分成 4 个对应的初始编码区间[0, 0.1)、[0.1, 0.5)、[0.5, 0.7)、[0.7, 1),如表 8.1 所示。

表 8.1 符号概率和编码区间关系示例

符号 S_i	S_0	S_1	S_2	S_3
符号概率 P_i	$P_0=0.1$	$P_1=0.4$	$P_2=0.2$	$P_3=0.3$
积累概率 P_{ci}	$P_{c0}=0.0$	$P_{c1}=0.1$	$P_{c2}=0.5$	$P_{c3}=0.7$
初始编码区间$[P_{ci}, P_{ci}+P_i)$	[0.0, 0.1)	[0.1, 0.5)	[0.5, 0.7)	[0.7, 1.0)

若输入符号序列$S_2S_0S_3S_0S_2S_3S_1$:对于第一个符号S_2,其编码区间为[0.5, 0.7);对于第二个符号S_0,初始编码区间为[0, 0.1),根据算法,应把相对于前一符号区间[0.5, 0.7)的[0, 0.1)作为新的区间(相当于前一区间的第一个 1/10 区间),即为[0.5, 0.52);依次类推。相当于根据符号在初始编码区间的位置,从[0, 1)中取出该段区间,然后扩展到[0, 1),供下一个符号在此基础上进行编码。具体编码过程如图 8.4 所示。

例 8.1 的解码过程实际上与编码过程是类似的。当接收到编码 0.5143876 后,判别其位于区间[0.5, 0.7),说明第一个符号为S_2;把S_2的区间进行扩展,编码相当于位于[0, 0.1)的区间,说明第二个符号为S_0;同理可得其他符号。

在算术编码中,常用小数的二进制表示法,例如:

$$3/8 = 0.375 = 0\times 2^{-1}+1\times 2^{-2}+1\times 2^{-3}$$

则 3/8 的二进制可以用系数表示为 011。

【例 8.2】 设信源字母表由 a、b、c 等 3 个符号组成,概率分别为 P(a)=1/2,P(b)=1/4,P(c)=1/4,则概率区间分别应为[0, 1/2)、[1/2, 3/4)、[3/4, 1)。

若输入符号序列为 abaca,编码过程见图 8.5。

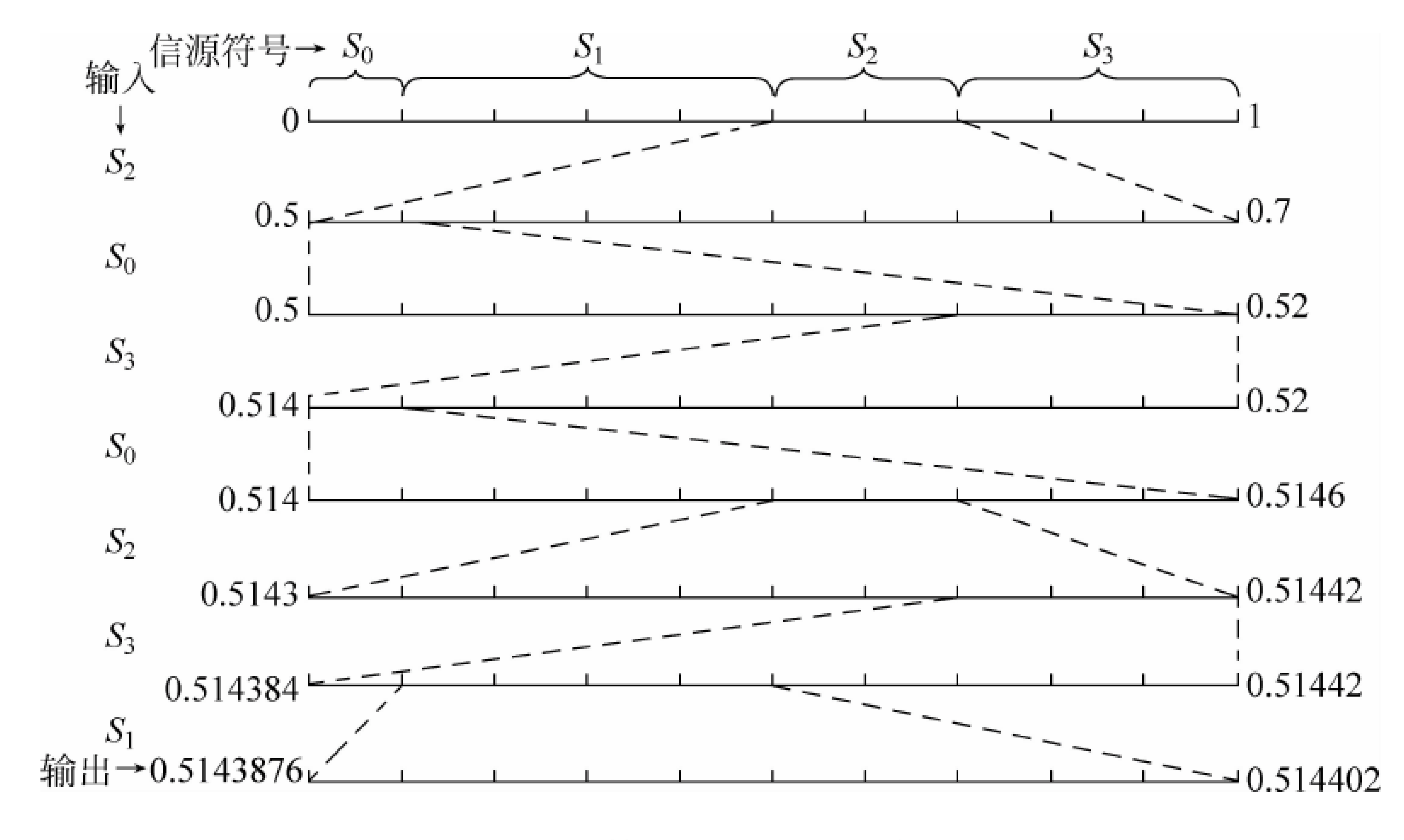


图 8.4　算术编码过程示例一

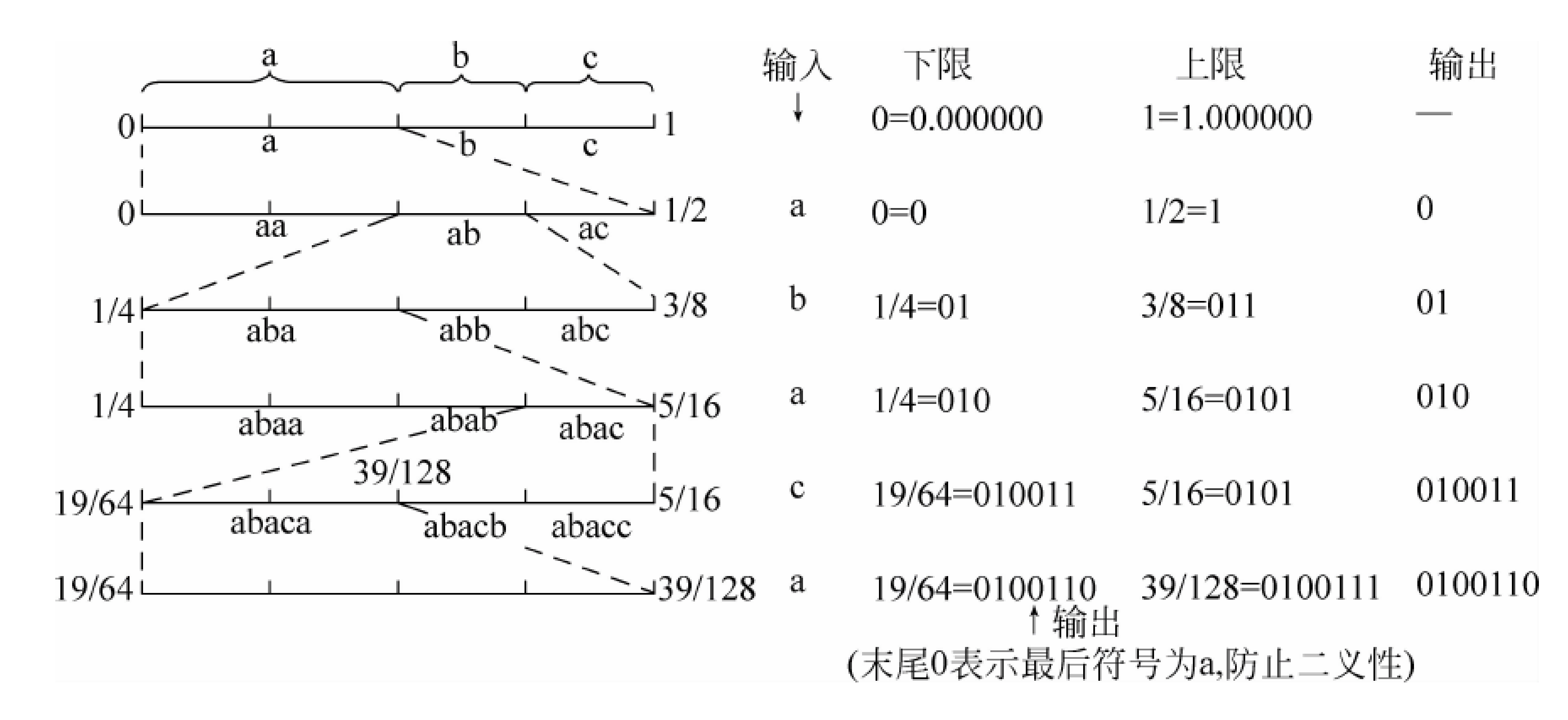


图 8.5　算术编码过程示例二

编码中有一个现象值得注意：序列 ab 和 aba 的编码都是 1/4＝01，存在二义性，所以，当最后一个字母为 a 时，可在编码后附加 0 来区分。

解码时，收到的第一个比特 0，位于区间[0，1/2)，可解出第一个字母为 a；收到第二个比特 1 后，得 01＝1/4，该区间位于前一区间的字母 b 位置，所以第二个字母为 b；依次解码后续字母。

在算术编码的实际运用中，最常用的方法是二进制算术编码，即先把输入数据转换为二进制符号串(二值化)，再进行算术编码。两个输入符号中，出现概率较大的称为 MPS，另一个称为 LPS，两者概率之和为 1。

算术编码可以有静态或自适应两种模式。静态算术编码的符号概率固定；自适应算术编码的符号概率是根据编码时上下文中符号出现频率进行动态调整，而在编码期间估计信源符号概率的过程就称为建模(Modeling)。由于符号概率对算术编码十分重要，但很难预知精确的信源概率，因此，动态建模就成为决定编码效率的关键。

算术编码对输入数据没有分组编码的要求，适用于自适应模式。因此，算术编码在 JPEG、JBIG、H.263 等标准中得到广泛应用，JPEG2000 中也使用了基于上下文的二进制算术编码方法。

8.2.4 ZIP 算法

ZIP 是一种非常常用的跨操作系统平台的数据压缩算法,在 Internet 上十分流行,例如,作为电子邮件的附件进行信息交换、进行文件下载等共享业务。数据压缩的好处既明显又直接,可以有效降低传输数据量,节省带宽和时间。

由于压缩对象主要是文档、程序等,显然,ZIP 必为无损压缩算法。ZIP 算法由 Jacob Ziv 和 Abraham Lempel 在 1977 年首次提出,因此也称 LZ77。

LZ77 算法的主要思想是基于滑动窗口缓冲区处理机制,并根据以下假设:一个文字流(或数据流)中的词或词组(或数据模式)有可能重复。这样,当所谓的重复发生时,重复部分将用较短的代码表示,数据量就会由此减少,达到“压缩”的目的。重复性越高,则压缩效果越好。

算法运行时,程序顺序扫描被压缩的文件,寻找“重复部分”,并同步地动态构造代码表,以便取代重复数据。代码还被用以捕获新的重复序列,当然,也被解压程序用以恢复原始数据。

先从一个简单例子着手考察 LZ77 算法的基本工作原理。设有字符串:

```
the brown fox jumped over the brown foxy jumping frog
```

包括空格在内,原始字符串长度为 53B,即 424b。

首先,将所有 8b 字符附加比特 1,转换为 9b 模式,以便解压程序了解凡是以比特 1 开头的 8b 编码为一个未经压缩的原始字符。

接着压缩程序从头开始扫描字符串,结果捕捉到两个重复串“the brown fox”和“␣jump”。

设计编码结构为 $0MPL$。其中二进制数值 0 用以区别于原始字符编码;M 为 1b 的编码类型,表示类型 0 和类型 1;P 为指针,指示重复了 P 个字符前的字符串;L 为重复串的长度。类型 0 的编码中,P 和 L 分别为 8b 和 4b;类型 1 的编码中,P 和 L 分别为 15b 和 6b,可用以表示隔得更远或长度更长的重复串。该编码结构也可表达为三元组 $<0x_{b}>$ $<P_{d}>$ $<L_{d}>$。

如图 8.6 所示,由于第二次出现的字符串“the brown fox”重复了 26 个字符前的字符串,长度为 13 个字符,则可采用编码类型 0 编码为 $<00_{b}>$ $<26_{d}>$ $<13_{d}>$。同理,重复字符串“␣jump”编码为 $<00_{b}>$ $<27_{d}>$ $<5_{d}>$。

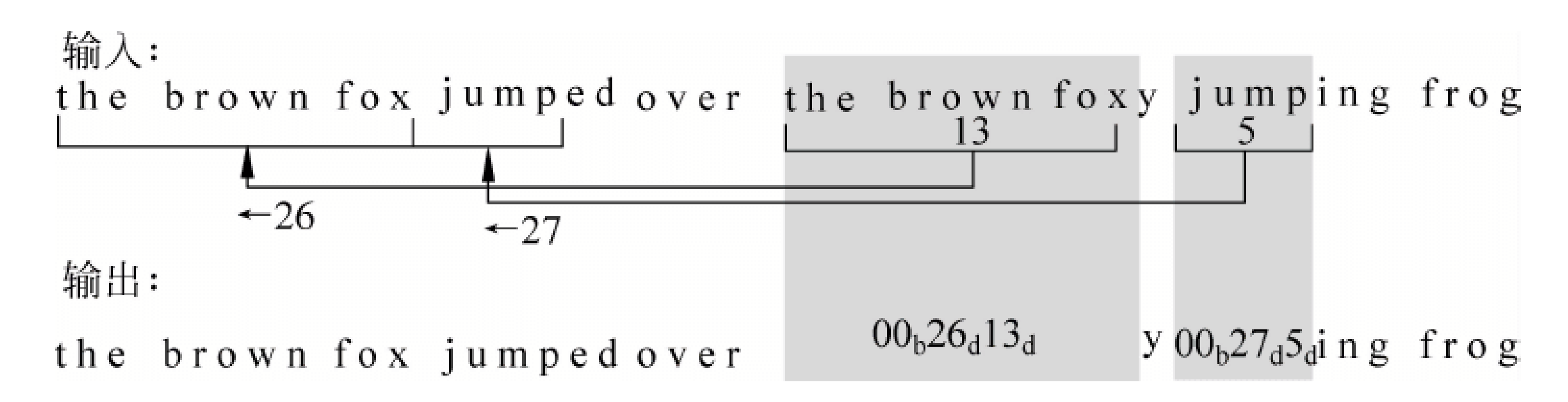


图 8.6 LZ77 数据压缩示例

算法输出的压缩数据为 $35\times(1+8)+2\times(2+8+4)=343\text{b}<424\text{b}$,压缩比为 $424:343\approx 1.24$,达到了数据压缩的目的。解压缩的过程非常简单,只需将 1 开头的 9b 编码去掉 1 后

还原为 8b 原始字符，若为 0 开头的编码，则依据编码三元组规则，还原出原始字符串。

LZ77 数据压缩算法及其变种采用双缓冲区扫描法。如图 8.7 所示，一个前视缓冲区依次装下 L 个待处理字符，另一个滑动历史缓冲区记录最后 N 个被处理的字符。

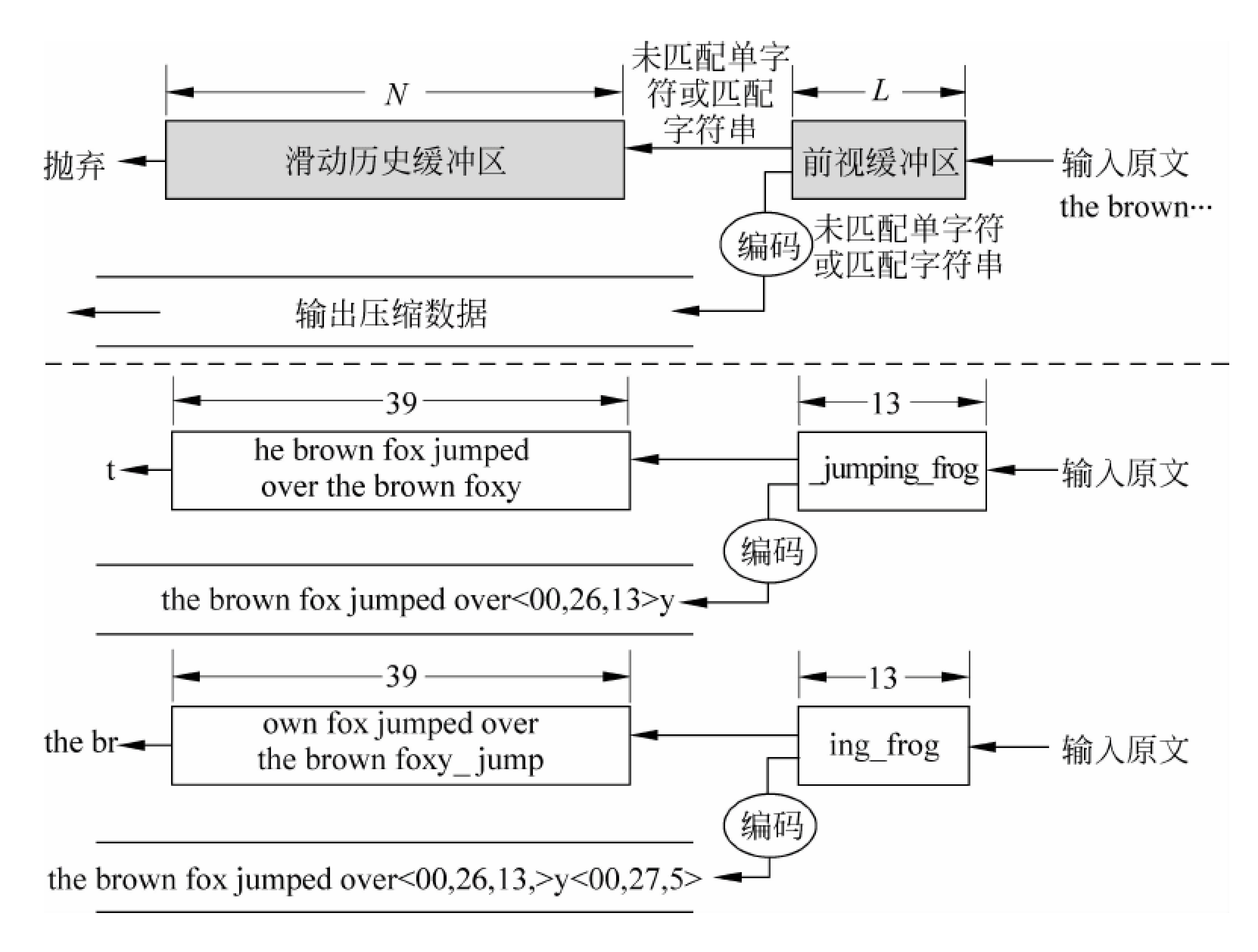


图 8.7　LZ77 双缓冲区扫描法及示例

算法按如下步骤执行。

(1) 从前视缓冲区的第一个字符开始，搜索在滑动历史缓冲区中出现的长度不低于 2 的匹配字符串。

(2) 如果找不到匹配字符串，则将前视缓冲区中的第一个字符变换为 9b 字符输出，同时该字符被推入滑动历史缓冲区，将滑动历史缓冲区中最老的字符挤出缓冲区。返回第(1)步。

(3) 如果找到了匹配字符串，算法将使匹配的字符串最长，并将该重复字符串用一个三元组编码取代。

(4) 如果重复字符串的长度为 K，则滑动历史缓冲区中最老的 K 个字符被挤出，刚被编码的 K 个字符被转入滑动历史缓冲区。

(5) 如果所有字符被处理完，算法结束，否则返回第(1)步。

LZ77 算法压缩速度快，有一定的自适应能力，但由于算法采用有限长度的窗口扫描文本，可能会因此遗漏潜在的、较长的重复串，从而降低压缩效率。虽然窗口可以被增大，但代价是搜索时间变长，同时指针编码的长度也要变长以满足长距离跳转的需要，有时候反而会降低压缩比。

思考：有哪些因素可影响 LZ77 的数据压缩比？LZ77 可以用于非字符型二进制数据的压缩吗？

8.2.5　离散余弦变换

离散余弦变换(Discrete Cosine Transform，DCT)是一种变换编码方法，用于图像的有

损压缩。变换编码的基本思路是:将原始信号的各个样值从通常的欧几里德几何空间(空间域)映射变换到另一个向量空间(变换域或频域),产生变换系数,再对变换系数进行编码。因为变换系数所需数据量远小于原始数据,可使编码后的码率得以显著降低。

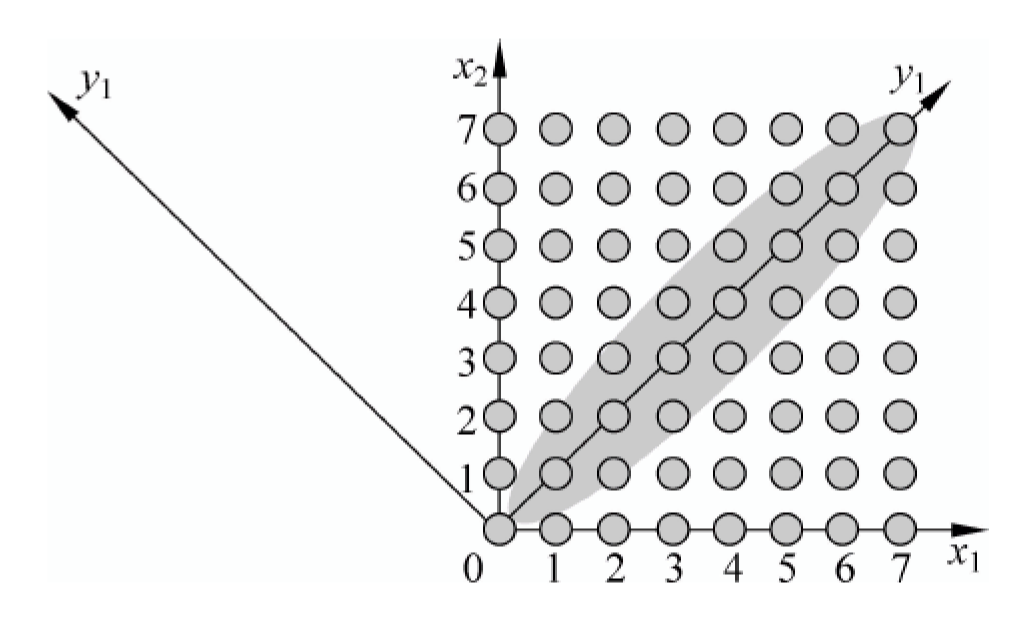


图 8.8 正交变换原理

信号的映射变换采用正交变换技术。如图 8.8 所示,设 x_1 和 x_2 是两个相邻的数据样本,每个样本有 $2^3=8$ 个幅度等级,可用 3b 编码。两个样本的联合事件共有 $8\times8=64$ 种可能性,可得以 x_1 和 x_2 为数轴的二维空间离散点的分布图。由于绝大部分声音、图像等信号,相邻样本较为近似,出现相近幅度等级的概率较大,即容易出现 $x_1=x_2$ 的情况,从而在图中表现为事件集中发生在 45°线附近阴影区的特征。

若对这组原始数据进行正交变换,相当于把坐标系旋转 45°,成为 y_1 和 y_2。可以发现,信息相关性越强,在 y_1 上表现为数值越大,而 y_2 上的数值越小,数据在 y_1 方向发生较大范围变化时,在 y_2 方向只有微小的变化,这意味着变量 y_1 和 y_2 在统计上具有相互独立性。因此,通过正交变换,能够使数据间的相关性大大减小,便于进一步的数据编码处理。

DCT 的实质正是通过线性变换 $\boldsymbol{X}=H(\boldsymbol{x})$,将一个 N 维向量 $\boldsymbol{x}$ 变换为变换系数向量 $\boldsymbol{X}$。H 称为 DCT 变换核,对第 k 行第 n 列的单个元素($k,n=0,1,\cdots,N-1$),定义

$$H(k,n)=c_k\sqrt{\frac{2}{N}}\cos\frac{(2n+1)k\pi}{2N}$$

其中 $c_0=\sqrt{2}$,$c_k=1(k>0)$。DCT 是线性正交变换,是可逆的。

在图像编码中,把图像分成独立的子块,以子块为单位进行二维 DCT 变换。二维 $M\times M$ 像素点的 DCT 变换公式为(设 $f(x,y)$ 为像素值)

$$F(u,v)=\frac{2}{M}C(u)C(v)\sum_{x=0}^{M-1}\sum_{y=0}^{M-1}f(x,y)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2M}$$

逆变换 IDCT 公式为

$$f(x,y)=\frac{2}{M}C(u)C(v)\sum_{u=0}^{M-1}\sum_{v=0}^{M-1}F(u,v)\cos\frac{(2x+1)u\pi}{2M}\cos\frac{(2y+1)v\pi}{2M}$$

其中(x,y)和(u,v)分别是图像子块和 DCT 空间坐标,可变系数 $C(0)=1/\sqrt{2}$,$C(i)=1(i>0)$。称 $F(0,0)=\frac{1}{M}\sum\sum f(x,y)$为直流系数,其他 $F(u,v)$为交流系数。

8.3 多媒体信息编码标准

8.3.1 字符编码

字符(文字)是信息表示的最重要、最基本的形式。

由于自然语言的多样性,字符的类型很多,例如中文(简体和繁体)、英文(大写和小写)、日文(平假名和片假名)等。除此以外,文本中还包括必不可少的空格、标点、符号,以及换行

符(CR/LF)、结束符(EOT)、制表符(TAB)、删除符(DEL/BS)等控制字符。一些特殊的控制字符(如标题开始、文本开始、转义符等)在早期的面向字符的电文发送系统中经常被通信规程用于简单的会话管理。

字符编码可为 5b、6b、7b 和 8b,汉字编码一般为 16b。Internet 上最常用的是 ASCII 码、Unicode 码(UTF-7 和 UTF-8),有些计算机采用 EBCDIC 码,中文编码标准有简体 GB 2312 和 GB 18030、繁体 Big5 等。

ASCII(American Standard Code for Information Interchange)编码和 **EBCDIC**(Extended Binary Coded Decimal Interchange Code)编码分别如表 8.2 和表 8.3 所示。

表 8.2 ASCII 码编码规则(7b)

L H	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	NL	VT	NP	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

表 8.3 EBCDIC 码编码规则(8b)

L H	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	NUL	SOH	STX	ETX	ST	TAB	SSA	DEL	EPA	RI	SS2	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	OSC	NEL	BS	ESA	CAN	EM	PU2	SS3	FS	GS	RS	US
2x	PAD	HOP	BPH	NBH	IND	LF	ETB	ESC	HTS	HTJ	VTS	PLD	PLU	ENQ	ACK	BEL
3x	DCS	PU1	SYN	STS	CCH	MW	SPA	EOT	SOS	SGCI	SCI	CSI	DC4	NAK	PM	SUB
4x	SP	NBSP	â	ä	à	á	ã	å	ç	ñ	[	.	<	(	+	!
5x	&	é	ê	ë	è	í	î	ï	ì	ß	]	$	*	)	;	^
6x	-	/	Â	Ä	À	Á	Ã	Å	Ç	Ñ	¦	,	%	_	>	?
7x	ø	É	Ê	Ë	È	Í	Î	Ï	Ì	`	:	#	@	'	=	"
8x	Ø	a	b	c	d	e	f	g	h	i	«	»	ð	ý	þ	±
9x	°	j	k	l	m	n	o	p	q	r	ª	º	æ	¸	Æ	¤
Ax	µ	~	s	t	u	v	w	x	y	z	¡	¿	Ð	Ý	þ	®
Bx	¢	£	¥	·	©	§	¶	¼	½	¾	¬	|	¯	¨	´	×
Cx	{	A	B	C	D	E	F	G	H	I	SHY	ô	ö	ò	ó	õ
Dx	}	J	K	L	M	N	O	P	Q	R	¹	û	ü	ù	ú	ÿ
Ex	\	÷	S	T	U	V	W	X	Y	Z	²	Ô	Ö	Ò	Ó	Õ
Fx	0	1	2	3	4	5	6	7	8	9	³	Û	Ü	Ù	Ù	APC

不同的计算机系统可能采用不同的编码标准,因此,在网络上进行数据交换,应约定并遵循同一编码规则。而实际的情况往往十分复杂,不仅限于端对端"双方认可"那么单纯,还

有多对多通信的需要,此外,网络系统中可能存在基于字符的通信设备或一些早期开发的网关、代理设备,需要考虑在各种条件下是否所有编码字符都能够透明穿越网络,而不会因为一些控制字符而对中继网络的通信造成影响。

Base64 就是一种电子邮件系统的常用编码技术,可以将任意二进制数据转换成可打印字符,从而最大限度地避免编码的二义性问题。

Base64 编码由 65 个可打印字符组成,其中"="用于填充(pad),其他 $2^6=64$ 个字符可以用 6b 来编码,如表 8.4 所示。

表 8.4　Base64 编码规则(6b)

H \ L	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1x	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f
2x	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
3x	w	x	y	z	0	1	2	3	4	5	6	7	8	9	+	/

如图 8.9 所示,Base64 先将任意 8b 数据划分为 6b 一组,再查表对应到 Base64 编码规则的可打印字符,这些字符可以进一步采用 ASCII 码或 EBCDIC 码进行编码并传输。显然,采用 Base64 编码将会增加数据量。

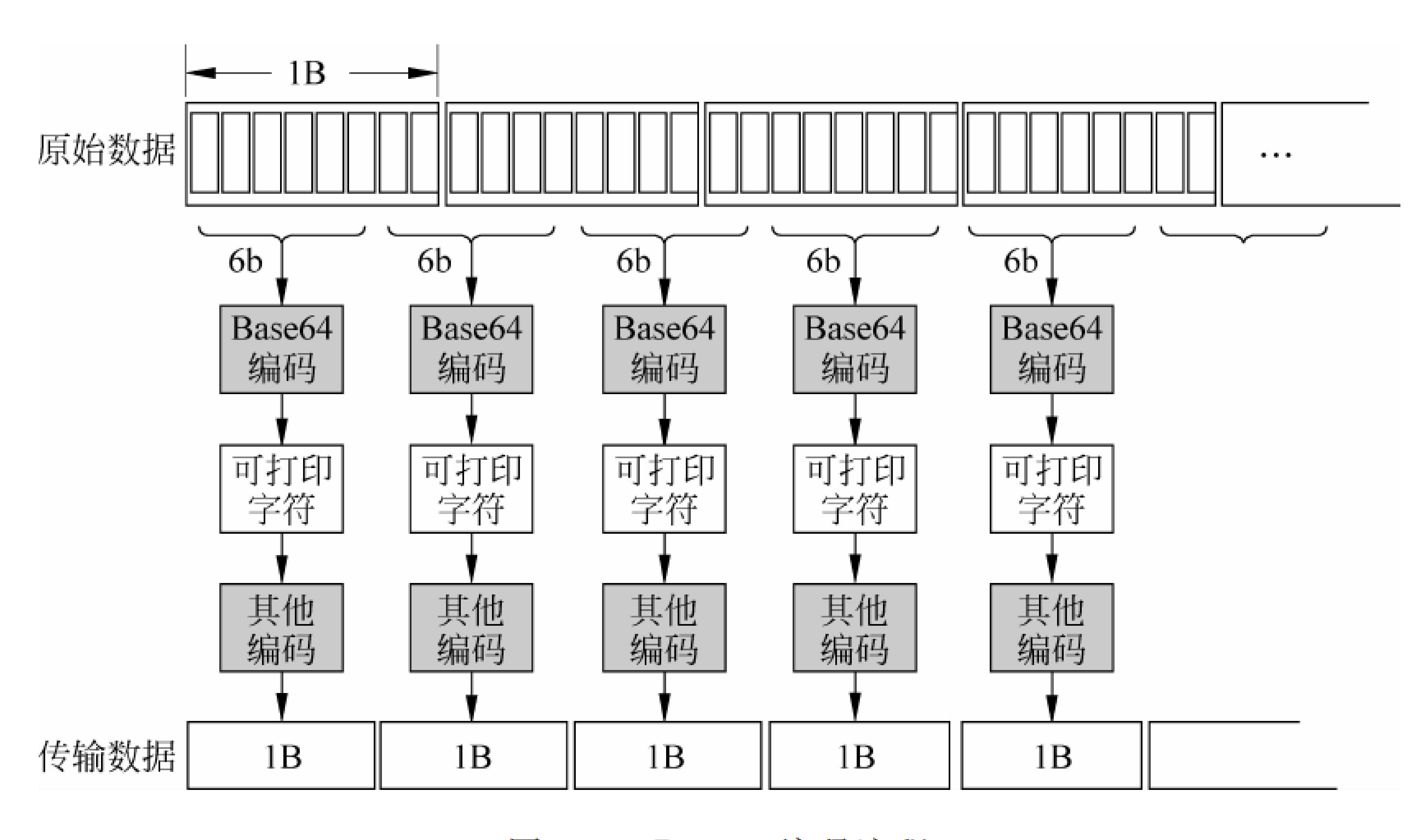


图 8.9　Base64 编码流程

例如,有 3B(24b)原始数据 00100011,01011100,10010001(0x23,0x5c,0x91),按 6b 一组划分为 001000,110101,110010,010001 共 4 组,查表产生 4 个可打印字符 I1yR,然后可采用一定的编码标准进行编码和传输,以 ASCII 码为例($bit_7=0$),编码后进行传输的数据为 01001001,00110001,01111001,01010010。接收方解码的方法为以上过程的逆过程。

实际上,在计算机网络诞生以前,字符编码技术就已经得到运用,例如著名的**莫尔斯电码**(Morse Code),主要应用于电报传输,非常适合当初技术简单的无线通信。虽然莫尔斯电码已经完全退出历史舞台,但了解其基本技术对于理解编码技术以及解读历史事件、阅读文学作品都有裨益。

莫尔斯电码采用长音“嘀”(可用划线“—”书写)和短音“嗒”(可用圆点“·”书写)的组合,表示各个字符,字符间用短暂停顿隔离,编码如表 8.5 所示。

表 8.5 莫尔斯电码编码

字符	电码	字符	电码	字符	电码	字符	电码	字符	电码
A	·—	B	—···	C	—·—·	D	—··	E	·
F	··—·	G	——·	H	····	I	··	J	·———
K	—·—	L	·—··	M	——	N	—·	O	———
P	·——·	Q	——·—	R	·—·	S	···	T	—
U	··—	V	···—	W	·——	X	—··—	Y	—·——
Z	——··	1	·————	2	··———	3	···——	4	····—
5	·····	6	—····	7	——···	8	———··	9	————·
0	—————	/	—··—·	+	·—·—·	=	—···—	.	·—·—·—
-	—···—	?	··——··	(	—·——·	)	—·——·—	,	——··——
"	·—··—·	'	·————·	:	———···	;	—·—·—·	$	···—··—

最有名的例子是在紧急的、特殊的情况下,可以发出“三短-三长-三短”(即嘀嘀嘀-嗒嗒嗒-嘀嘀嘀)信号,用以传递出国际通行的“SOS”求救信息。

8.3.2 静态图像编码

静态图像是多媒体信息中最常见的信息类型之一。一幅数字化的图像是由矩阵式像素点构成,每个像素点的属性可采用不同的方法来描述。非彩色图像是由白色、不同等级的灰色和黑色组成,对光谱各波长的反射没有选择性,是中性颜色;彩色图像(物体)则对光谱具有波长选择性。

一个能发出光波的物体为有源物体,颜色由物体发出的光波决定,通常是由三基色红(Red)、绿(Green)和蓝(Blue)按不同比例合成,构成一个三维的 RGB 矢量空间,并被称为相加混色模型;而不发光的物体为无源物体,其颜色由物体吸收或者反射的光波决定,是由青(Cyan)、品红(Magenta)和黄(Yellow)按不同比例合成(例如颜料的调配),称为 CMY 相减混色模型,增加单独的黑色(Black)后也称 CMYK。CMY 颜色模型常用于印刷、打印领域,在数字图像处理中较少用到。

颜色还具有亮度(Luminance)、色调(Hue)和饱和度(Saturation)3 个特性,据此来描述颜色的模型称为 HSL 模型。亮度是颜色的明暗程度,与色光所含能量有关;色调指颜色的类别,由光谱分布决定;饱和度是指色调深浅的程度(俗称色彩是否鲜艳),单色光中掺入的白光越多则饱和度越低。色调和饱和度合称为色度(Chromaticity)。

由于人眼视觉系统对色度信号的敏感程度不如亮度信号,因此将亮度和色度独立表示是降低数据量的一种途径。例如 YUV 表示法,Y 是亮度信号,Y 信号分量构成黑白灰度图,与用 U、V 信号构成的另外两幅单色图相互独立,分别编码。例如对 RGB 的 8∶8∶8 彩色图像(RGB 分量都用 8b 表示),640×480 像素大小需要 921 600B,而用 YUV 表示,Y 分量为 8b,而对 2×2 的 4 个相邻像素的 U、V 值用一个值表示,可将数据量减少到 460 800B,不失为一种有效的图像压缩技术。

与 YUV 类似的还有 YIQ 和 YCrCb,分别在 PAL、SECAM、NTSC 制式的彩色电视机

和计算机显示器中采用。

RGB 与 YUV、YCrCb 色彩空间的变换关系如下：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} Y \\ \mathrm{Cr} \\ \mathrm{Cb} \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.500 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

静态图像的压缩标准很多，如 BMP(Bit-Map Picture，位图图片)、GIF(Graphics Interchange Format，图形交换格式)、PNG(Portable Network Graphics，可移植性网络图像)、TIFF(Tag Image File Format，标签图像文件格式)，而最为成功、应用最广的是 JPEG (Joint Photogragh-coding Experts Group，**联合图像编码专家组**)标准。

JPEG 有无损压缩和有损压缩两种方式，无损方式压缩比较低，有损方式可提供不同压缩比的选择，压缩比越高，失真度越大。JPEG 具有适中的计算复杂性，有利于软件和硬件实现。JPEG 标准规定了三种编码系统：

(1) 基于 DCT 的有损编码基本系统，适用于大多数场合；

(2) 用于高压缩比、高精度或渐进重建应用的扩展编码系统；

(3) 用于无失真应用场合的无损系统。

JPEG 能够处理 4b/pixel 到 16b/pixel 的图像。JPEG 编码过程包括以下主要步骤。

(1) 色彩空间转换。从 RGB 转换为 YCrCb。

(2) 零偏置转换。对于灰度级是 2^n 的像素，像素值减去 2^{n-1}，使值域转换为 -2^{n-1} ～ $2^{n-1}-1$，使绝对值变小，减少位数，有利于熵编码。

(3) 把图像分割成互不重叠的 8×8 子图。

(4) 对每个子图进行 DCT 变换。

(5) 对变换系数进行量化。

(6) 对量化系数进行熵编码得到压缩码流。

JPEG 的解码是该过程的逆过程。

JPEG2000 标准是对 JPEG 的发展，但改用**离散小波变换**(Discrete Wavelet Transform，DWT)为主的多解析编码方式。

8.3.3 音频编码

ITU-T 制定的 G 系列标准，如 G.711、G.721、G.722、G.728、G.729，是 PSTN/ISDN 中采用的音频编码标准，码率在 8～64Kb/s 之间。

ISO 与 ITU-T 联合制定的 MPEG(Moving Pictures Experts Group，**动态图像专家组**)标准在计算机网络领域得到广泛应用。

MPEG-1 的音频编码提供 3 个独立层次：MP1、MP2 和 MP3。MP1 编码器最简单，输出码率为 384Kb/s，用于小型数字盒式磁带；MP2 的编码器复杂度中等，输出码率为 192～256Kb/s，应用包括数字广播音频、数字音乐、CD-I 和 VCD；MP3 编码器最复杂，输出码率为 8～128Kb/s，主要用于音乐文件存储和网络声音传输，在 Internet 上得到普遍应用。

MPEG-2 音频编码提高了低采样率下的声音质量，支持多通道环绕立体声和多语言技术，包括 BC 和 AAC(Advanced Audio Coding)两种算法，后者支持 1～48 个通道，每通道可获得 8～160Kb/s 高质量声音。MPEG-2 可提供较大的可变压缩比，以适应不同声音质量、存储容量和传输带宽的要求，适用于数字音频广播、DVD、多声道数字电视声音和多媒体网络传输。

MPEG-4 音频标准可集成从话音到高质量的多通道声音，不仅支持自然声音，还支持合成声音(如文本-语音转换 TTS)。MPEG-4 采用可变速率编码器，输出速率 200～64Kb/s，有较强的应用灵活性。

8.3.4　视频编码

数字视频的数据量非常庞大。标准清晰度的 NTSC 视频为 30fps、4∶2∶2YCrCb、720×480，要求超过 165Mb/s 的传输速率，如果不进行有效的压缩，几乎没有应用价值。

视频编码除了需要满足压缩数据量的需要，还需要同时考虑声音(如伴音)的编码，并在播放时保持视频和音频的同步。主流的视频编码标准有：

(1) 基于块的混合视频编码方案，有 H. 261、H. 263、H. 264、MPEG-1、MPEG-2、MPEG-4、AVS 等标准，采用预测编码、变换编码、量化编码和熵编码技术；

(2) 基于小波变换的视频编码方案，实现可伸缩的视频编码；

(3) 基于内容和对象的视频编码方案。

基于块的视频编码是最成熟、实用性最强、使用最广泛的技术。其基本技术思想有两点：在空间方向上，采用类似于 JPEG 的压缩算法去除空间冗余信息；在时间方向上，采用预测编码去除时间冗余信息。

视频压缩将帧分为三种类型：**I 帧**(Intra Frame)、**P 帧**(Predicted Frame)和 **B 帧**(Bi-directional Frame)。如图 8.10 所示，I 帧不依赖于其他帧，而是靠尽可能去除帧内空间冗余来实现数据压缩的编码帧，因此称为帧内编码帧；P 帧是通过重复利用视频序列中之前已经完成的编码帧的时间相关性来消除时间冗余信息，称为预测帧；B 帧既利用视频序列前面已编码的帧，又利用视频序列后面已编码帧的相关性，以消除时间冗余信息，所以称为双向预测帧。一般 I 帧压缩率最低，P 帧其次，B 帧最高。P 帧通常使用前面最近的 I 帧或 P 帧作为参考图像，B 帧使用前后两帧 I 帧或 P 帧作为预测参考，其中一个参考帧在显示顺序上先于编码帧，是前向预测，另一参考帧属于后向预测。

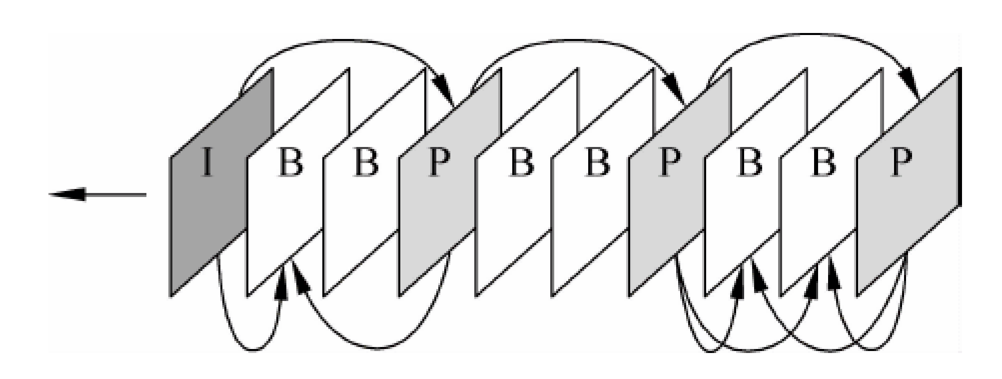


图 8.10　视频编码的基本帧类型

通常把连续的帧分为图像组(Group Of Pictures，GOP)，GOP 的第一帧为 I 帧，其余为 P 帧和 B 帧。

预测编码是一种基于运动补偿(motion compensation)方法的技术。首先应进行运动估计(motion estimation)，以块匹配算法(Block-Match Algorithm，BMA)为例，如图 8.11

所示,将图像分成 $N \times N$ 的**宏块**(macro block),用当前图像的每一个宏块在上一帧的一定范围(匹配窗)内搜索与之最接近的预测块,得到预测块到当前块的位移,就是**运动矢量**(motion vector)。将前一帧相应的运动部分信息根据运动矢量补偿过来的过程就是运动补偿。而预测块与当前块之间的差值称为残差图像,因此,每个当前图像宏块都可以用残差图像和一个运动矢量来表示。预测越准,残差中的数值越小,编码后占用的比特数就越少。

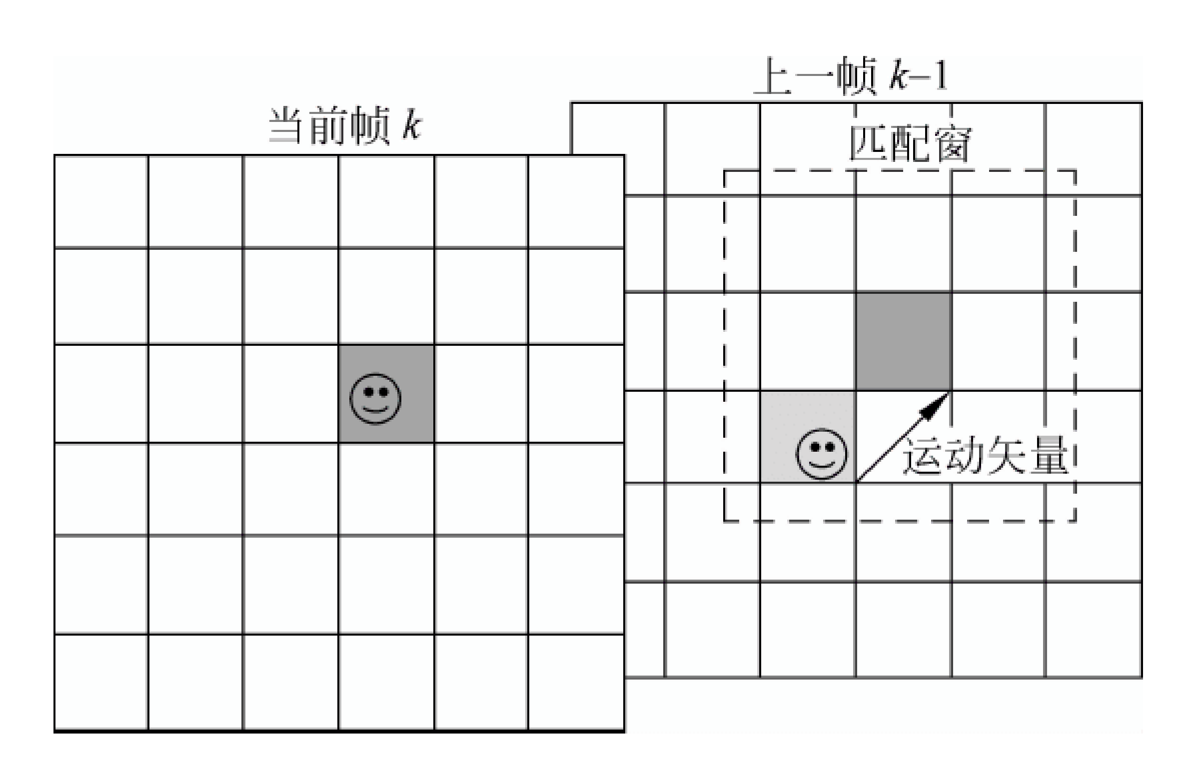


图 8.11 预测编码宏块匹配

宏块的大小应是一个折中的选择:若太小,可能匹配到错误的块,且增加运算量,同时增加所需传输的运动矢量信息;若太大,可能块中存在不同的运动矢量,就难以提供准确、有效的估计。

主要的视频编码标准技术指标对比如表 8.6 所示。

表 8.6 主要的视频编码标准技术指标对比

视频编码标准	像素	fps	b/s	质量	应 用
MPEG-1	352×288	25/30	1M～1.5M	CIF	VCD、视频点播
MPEG-2/H.262	720×486	30	3M～10M	PAL/NTSC	DVD、视频广播/点播
H.261	352×288	30	64K～2M	QCIF	视频会议、可视电话
H.263	可变	30	64K	QCIF/CIF	视频会议、可视电话
MPEG-4/H.264	质量高、码率低、可调性好、适应性强				网络视频、视频广播

8.3.5 数字水印

Internet 的繁荣对知识产权而言似乎是一场愈演愈烈的灾难,因为便捷的访问、轻松的共享、快速的交互加上缺位的监管,使侵犯版权、著作权的事件层出不穷,所以,面向数字化产品的**数字版权保护**(Digital Right Protection,DRP)显得十分迫切。但在虚拟世界中,这无疑是一项艰巨的任务。

数字水印(Digital Watermark)是一种对图像、视频、音频等数字作品提供版权认证的技术。数字水印的基本原理是:将代表数字媒体著作权人身份的特定信息、标志或序列码等信息,按照某种方式嵌入被保护的信息中,在需要时可通过相应的算法提取该水印,用以验证版权的归属。当然,数字水印的加入必须以不破坏被保护对象的使用价值、欣赏价值为前提。

数字水印通常为不可感知型的,除此之外,还具备如下特点。

(1) 嵌入有效性。应保证嵌入的水印能够被成功检测到。

(2) 抵抗破坏的鲁棒性。当图像进行裁剪、扭曲、有损压缩,甚至打印和扫描后,仍能够检测或部分检测到水印的存在。但有些用于认证的水印故意设计为脆弱的。

(3) 避免虚检水印。水印检测算法不能够在不含水印的对象中提取到虚假的水印。

(4) 水印安全性。防范攻击者破译水印算法并篡改水印信息。

空域数字水印算法是最早出现也是比较简单的方法,其工作原理是在空间域中按某种规则直接修改原始载体数据,加载水印信息。信息将被嵌入到随机选择的**最不重要像素位**(Least Significant Bit,LSB)上,使水印不可见。但正是由于位于 LSB 上,水印信息很容易被滤波、量化、变形等操作破坏。

另一种 Patchwork 算法是随机选择 N 对像素点$\{a_i,b_i\}$,将每个 a_i 点的亮度值加 1,每个 b_i 点的亮度值减 1,保持整个图像的平均亮度不变。提取时,同样选择这 N 对像素点$\{a'_i,b'_i\}$,计算 $S=\sum(a'_i-b'_i)$,若 S 接近 $2N$,说明含有水印,如果趋近于 0,说明不含水印。Patchwork 算法对 JPEG 压缩、FIR 滤波及图像裁剪有一定的抵抗力,但嵌入的信息量比较有限。

其他数字水印技术有 DCT 域数字水印、DFT 域数字水印、小波域数字水印等。另外还有专门面向视频、音频的数字水印生成和检测技术。

数字水印本身的安全性也面临着诸多挑战。有一种称为马赛克的扰乱攻击,将图像分为许多特别小的图像,以至于每个都无法进行可靠的水印检测,而拼接后的显示不受任何影响;通过一个信号扰乱器可扰乱录像机的输入,使录像机检测不到水印而准许录像,播放时用相应的解扰器恢复信号,巧妙绕过了防复制机制;利用水印实现技术对同步非常敏感的特点,通过干扰同步来掩蔽水印信号;采用线性低通滤波器,削弱高频部分具有高能量的水印,使水印无法检测;复制攻击则将水印从一个载体复制到另一载体,可用以伪造名家作品的赝品,达到以假乱真、鱼目混珠的目的。

CHAPTER 9

第9章 密码学基础

9.1 信息加密原理

信息加密(Information Encryption)作为保障数据(消息)安全的一种方式,其历史相当久远,可能要追溯到公元前2000年。虽然那个时代的密码与现代加密技术不可同日而语,但已具备加密的概念和雏形,也说明了信息加密的重要性。

如图9.1所示,信息加密的基本原理是:把**明文**(plaintext)用**加密**(encryption)方法和**密钥**(key)生成保密的**密文**(cryptograph或ciphertext),只有使用正确的**解密**(decryption)方法和密钥后才能还原出明文。

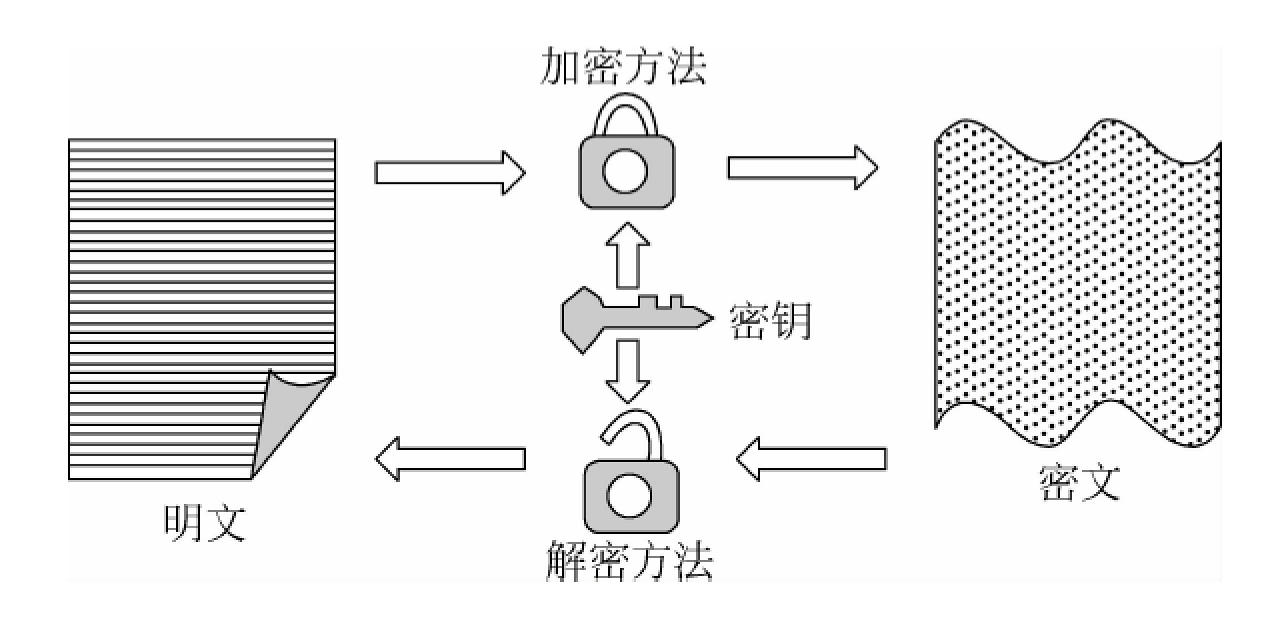


图9.1 加密和解密基本原理

设明文空间为T,密钥空间为K,密文空间为C,E为加密方法,D为解密方法。若有明文$t \in T$,加密密钥为$k_e \in K$,解密密钥为$k_d \in K$,密文$c \in C$,则加密和解密过程可分别表示为

$$c = E(k_e, t)$$

$$t = D(k_d, c)$$

一个完善的密码体制至少应满足两个条件:

(1)已知明文t和加密密钥k_e时,容易计算$c=E(k_e,t)$和$t=D(k_d,c)$;

(2)在不知道解密密钥k_d时,难以由密文c推知明文t。

信息加密可以实现**信息保密性**(防止用户的标识或数据被非法窃取)、**数据完整性**(防止数据被非法篡改和伪造)、**不可否认性**(确定数据来源的可靠性,并防止抵赖)。

然而,没有绝对不可破译的加密方法。例如可以采用穷举法尝试所有的可能性。保密的相对性主要是指计算上的,就是利用现有最高性能的计算机是否能在信息失效前实现破解。而且,加密是一把双刃剑,复杂的加密、解密算法将消耗大量的系统资源、增加系统负荷,反过来会影响信息系统正常运行的效率。所以,设计和应用加密方法应讲究策略,选择适合需求的方案,使破译的成本超过被加密信息的价值,或使破译的时间超过被保护信息的有效期。

加密技术可分为**传统加密技术**(古典密码)和**现代加密技术**。

传统加密技术主要采用置换方法,例如改变字母的映射关系、对字母进行编码或替换等。因为置换方法不改变字母出现的频率,所以运用统计学上的频率分析方法就能破译密文(如图 9.2 所示)。

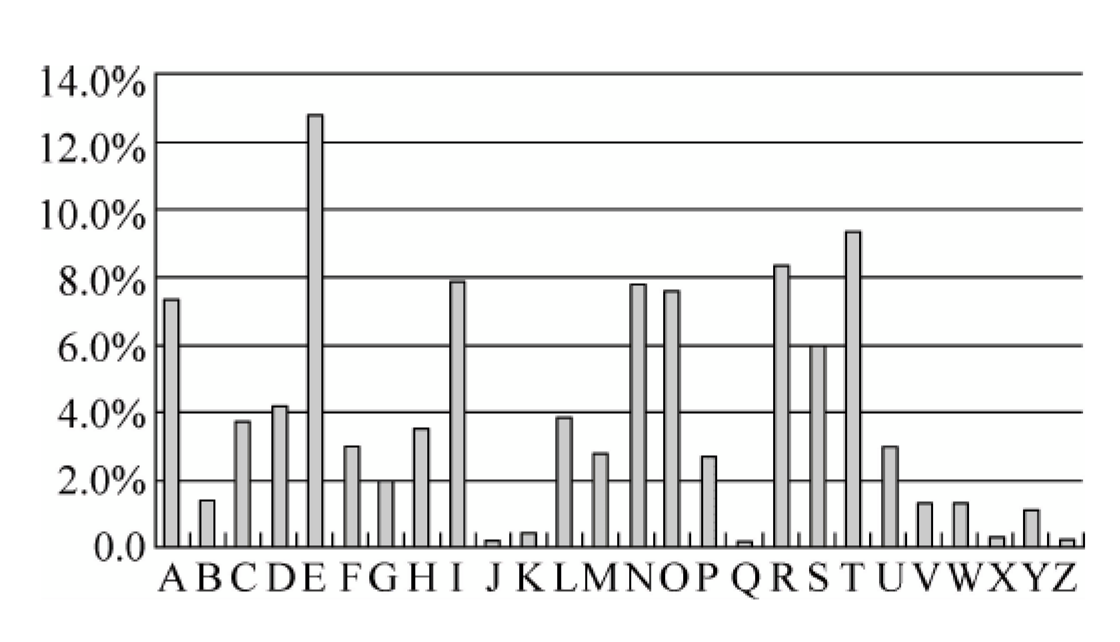


图 9.2　英文字母频率统计

现代加密技术包括对称密钥加密、非对称密钥加密、单向函数加密三种类型,以数学方法为基础,保密强度较高,并体现出如下特征。

(1) 以加密(解密)算法为核心。算法的研制是最重要、最根本的技术,同时也是十分困难的、富有挑战性的工作。加密和解密算法可以公开,事实上,通常成为国际(国家)标准。

(2) 加密强度主要依赖于密钥,具体地,依赖于密钥的长度,所以应用中非常强调密钥的管理。

(3) 加密算法便于计算机实现(甚至采用 IC 芯片实现)和网络应用。

9.2　古典密码

9.2.1　Greece 密码

希腊(Greece)密码出现于公元前 2 世纪,是一种二维字母表编码方法。如图 9.3 所示,将 26 个字母分为 5×5 的矩阵,用行列号作为对应字母的编码。

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I/J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

图 9.3　Greece 密码编码表

例如,明文 FUDAN 可编码加密为 21 45 14 11 33。设想一位古代将军,拿到截获的敌方机密文件,看到的却是一连串数字,一定会一头雾水、非常烦恼。而拥有同一张(同

一规则)编码表的人却很容易就能解密得到明文。

字母表还可以变化为4×7或3×8(再合并U和V)。

9.2.2 Caesar 密码

凯撒(Caesar)密码是一种古老的单表置换密码。方法是把字母表中的每个字母用该字母后面第 n 个字母进行替换。如果 $n=3$,则变换表如下。

字母表:a b c d e f g h i j k l m n o p q r s t u v w x y z

变换表:d e f g h i j k l m n o p q r s t u v w x y z a b c

例如,明文为 meet me after the party,密文为 phhwphdiwhuwkhsduwb。

如果对26个英文字母依次进行赋值1,2,…,26,设 t 为明文字母,c 为密文字母,$k=1,2,\cdots,25$ 为步长(即密钥),则 Caesar 密码用数学算式表达为

加密算法:$c=\mathrm{E}(t)=(t+k)\bmod 26$

解密算法:$t=\mathrm{D}(c)=(c-k)\bmod 26$

显然,假如已知加密用的是凯撒密码,则使用暴力攻击(穷举法)密码分析最多只需25次尝试。

9.2.3 Prefix 密码

标准字头(Prefix)密码又称为密钥短语密码,属于单表置换密码,与凯撒密码类似,但利用一个密钥字来做表头,构造变换表。密钥字通常是单词或词组,便于记忆(例如可以把密钥字刊登在报纸的约定位置上)。

设密钥字 $K=$cat,获得变换表如下。

字母表:a b c d e f g h i j k l m n o p q r s t u v w x y z

变换表:c a t b d e f g h i j k l m n o p q r s u v w x y z

思考:为什么不要选用 cab 作为密钥字?如何选取密钥字可使发生变化的字母最多?

该方法还可以做一种变形:选取一个字母,然后从该字母位置开始配置变化表。例如选取 e,则变换表中 c 应该从字母表的 e 位置开始,再依次往后。

再例如密钥字为 HAPPY NEW YEAR,可先去掉其中重复的字母得到一个无重复字母的字母串,即 HAPYNEWR,而后依法生成变换表。

9.2.4 Playfair 密码

公平竞赛(Playfair)密码属于多表加密体制,将明文中的双字母组合作为一个单元对待,并将这些单元转换为密文双字母组合。

Playfair 算法使用一个5×5字母矩阵,该矩阵使用一个关键词构造。

例如,关键词是 monarchy,从左至右、从上至下填入该关键词的字母(去除重复字母),然后再按字母表顺序将余下的字母填入矩阵剩余空间,字母 I 和 J 被算作一个字母,由此得到 Playfair 字母矩阵为

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Playfair 根据下列规则依次对明文的两个字母加密。

(1) 属于相同对中的重复的明文字母将用一个填充字母(例如 x)进行分隔。例如，balloon 会出现 ll，则将被改为 ba lx lo on。

(2) 属于矩阵相同行的明文字母将由其右边的字母代替，而行的最后一个字母由行的第一个字母代替。例如，ar 被加密为 RM。

(3) 属于矩阵相同列的明文字母将由它下面的字母代替，而列的最后一个字母由列的第一个字母代替。例如，mu 被加密为 CM。

(4) 明文的其他字母将由与其同行，且与下一个字母同列的字母代替。例如，hs 成为 BP，ea 成为 IM(或 JM，可根据加密者的意愿而定)。

规则(4)等价于在字母矩阵中以两个明文字母为顶角画矩形，矩形的另一对顶角即为密文字母对。规则(2)和(3)只是处理明文字母对在同行和同列时的特殊情况。

Playfair 密码与简单的单一字母替代法密码相比有了很大的进步。第一，虽然仅有 26 个字母，但有 26×26=676 种双字母组合，因此，识别各种组合要困难得多。第二，频率分布不规则，使得频率分析困难得多。由于这些原因，Playfair 密码过去长期被认为是不可破的，曾被英国陆军在第一次世界大战中作为顶级加密系统使用，在第二次世界大战中仍被美国陆军和其他同盟国大量使用。

9.2.5　Vigenere 密码

维吉尼亚(Vigenere)密码最早记录在吉奥万·巴蒂斯塔·贝拉索(Giovan Battista Bellaso)1553 年的密码技术著作中，19 世纪时被误传为法国外交官布莱斯·德·维吉尼亚(Blaise De Vigenere)所发明，因此现在被称为 Vigenere 密码。

Vigenere 密码是一种非常简单，也非常著名、非常有效的单字符多表替换密码。其加密方法是：

为了加密一个消息，需要一个与该消息一样长的密钥。通常该密钥为一重复的关键词(密钥字)。设 26 个字母依次赋值为 0，…，25。

设密钥 $K=k_1k_2\cdots k_n$，明文 $M=m_1m_2\cdots m_n$，则 $E_k(M)=c_1c_2\cdots c_n$，其中 $c_i=(m_i+k_i)$ mod 26。

另一种加密方法为事先构造一个 Vigenere 矩阵，通过查表来加密信息。Vigenere 矩阵由步长为 0～25 的 26 行凯撒密码序列构成。若给定一个密钥字母 d 和一个明文字母 w，密文字母就位于(d, w)上，即密文为字母 z。

```
abcdefghijklmnopqrstuvwxyz
bcdefghijklmnopqrstuvwxyza
cdefghijklmnopqrstuvwxyzab
defghijklmnopqrstuvwxyzabc
efghijklmnopqrstuvwxyzabcd
…
zabcdefghijklmnopqrstuvwxy
```

例如，设关键词是 deceptive，明文消息为“we are discovered save yourself”，查表加密结果如下。

密钥：d e c e p t i v e d e c e p t i v e d e c e p t i v e

明文：w e a r e d i s c o v e r e d s a v e y o u r s e l f

密文：z i c v t w q n g r z g v t w a v z h c q y g l m g j

解密方法同样简单：由密钥字母标识行，该行中密文字母所在位置决定列，明文字母位于该列的顶部。

Vigenere 密码的强度在于每个明文字母由多个密文字母对应，每个明文字母可以对应于该关键词的各个独特的字母，因此，该字母的频率信息变得模糊了。然而，并非所有明文结构的所有知识都丢失了。对于具有长度为 9 的关键词的 Vigenere 密码的频率分布，虽然取得了优于 Playfair 密码的改进，但仍然保留了可观的频率信息。

与 Vigenere 密码相似的有 Beaufort 密码。

9.2.6 Vernam 密码

弗纳姆(Vernam)密码是 1918 年 AT&T 工程师 Gilbert Verna 设计的，创新性地用二进制数据而不是用字母工作。

Vernam 加密体制可简单地表示为

$$c_i = p_i \oplus k_i$$

式中：p_i 为明文的第 i 个二进制数字；k_i 为密钥的第 i 个二进制数字；c_i 为密文的第 i 个二进制数字；$\oplus$为异或操作。

通过执行明文和密钥的逐位异或操作，产生密文。因异或的性质，解密仅需执行相同的逐位异或操作：

$$p_i = c_i \oplus k_i$$

Vernam 密码技术的核心是密钥构造的方法。Vernam 提议使用循环的带子，允许使用很长的密钥。尽管长密钥使得密钥分析极为困难，然而，只需找到充足的密文，使用已知的或可能的明文序列，通过两者的结合、比对就能够帮助破译。

Joseph Mauborgne 提出了对 Vernam 密码的改进方案，使用一种恰好与消息一样长度的随机序列密钥，该密钥没有重复。这种方案被称为“一次一密”，号称永远不可破译，因为其输出的密文与明文没有任何统计关系，还因为当使用暴力法破译时，可以找到大量可能的候选“明文”，产生多对一的关系，使穷举式破译过程无法停止(如图 9.4 所示)。然而，这种方法带来另一方面的困扰，发送者和合法接收者必须传递、保存并保护该随机密钥，这个过程往往会产生很大的问题。因此，虽然这一改进在加密技术中性能卓越，但很少被使用。

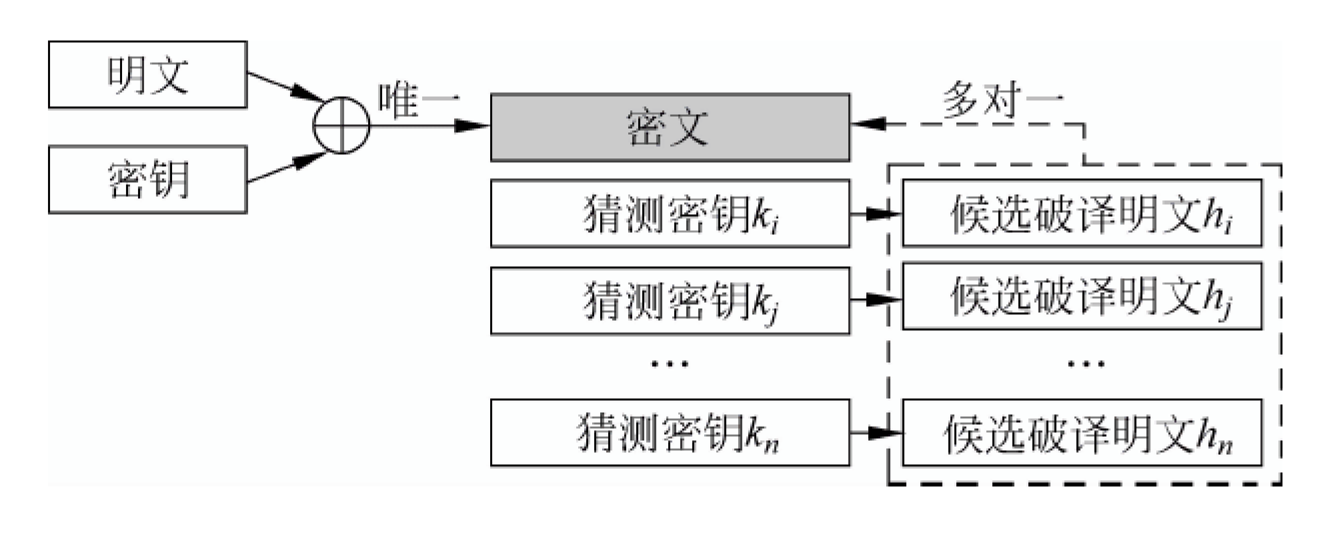


图 9.4 一次一密法加密和解密分析

9.2.7 Hill 密码

希尔(Hill)密码是一种有趣的多字母密码加密方法，由数学家 Lester S. Hill 于 1929 年研制，其原理是矩阵的线性变换。

Hill 密码算法取 m 个连续的明文字母，并用 m 个密文字母代替。这种替代由 m 个线性方程决定，其中每个字符被分配一个数值($a=0,b=1,\cdots,z=25$)。若 $m=3$，该系统可以用方程组描述如下：

$$\begin{cases} C_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26 \\ C_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26 \\ C_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26 \end{cases}$$

用向量矩阵乘法表示为

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

即 $\boldsymbol{C}=\boldsymbol{K}\cdot\boldsymbol{P}$。其中：$\boldsymbol{C}$ 和 $\boldsymbol{P}$ 是长度为 3 的列向量，分别表示明文和密文；$\boldsymbol{K}$ 是一个 3×3 矩阵，表示加密密钥。操作要执行模 26 运算。

例如，明文为“pay more money”，设加密密钥为

$$\boldsymbol{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

明文的前三个字母被表示为向量形式，则有

$$\begin{pmatrix} p \\ a \\ y \end{pmatrix} \rightarrow \boldsymbol{K} \times \begin{pmatrix} 15 \\ 0 \\ 24 \end{pmatrix} \rightarrow \begin{pmatrix} 375 \\ 819 \\ 486 \end{pmatrix} \bmod 26 \rightarrow \begin{pmatrix} 11 \\ 13 \\ 18 \end{pmatrix} \rightarrow \begin{pmatrix} L \\ N \\ S \end{pmatrix}$$

同理，以三个字母一组的方式继续计算下去(忽略空格)，遍历所有的明文字母，最后得到密文为 LNS HDL EWM TRW。

解密过程则使用矩阵 $\boldsymbol{K}$ 的逆矩阵 $\boldsymbol{K}^{-1}$，有 $\boldsymbol{P}=\boldsymbol{K}^{-1}\cdot\boldsymbol{C}$。其中求逆矩阵 $\boldsymbol{K}^{-1}$ 可利用矩阵 $\boldsymbol{K}$ 的伴随矩阵 $\boldsymbol{K}*$ 和矩阵的行列式 $|\boldsymbol{K}|$，关系式如下：

$$\boldsymbol{K}^{-1} = \frac{\boldsymbol{K}^*}{|\boldsymbol{K}|} \bmod 26$$

与 Playfair 算法相比，Hill 密码的保密强度在于其可以完全隐藏单字母的统计频率。一个 3×3 的 Hill 密码不仅隐藏了单个字母，而且也隐藏了两个字母的频率信息。如果使用较大的矩阵就可以隐藏更多的频率信息。

尽管 Hill 密码对抗“仅有密文攻击”的强度较高，但也比较容易被“已知明文攻击”所攻破。

9.2.8 Enigma 密码

Enigma 密码又称转子机密码，是二次大战中使用的最著名的加密技术之一。Enigma 的字面意思就是“谜”，该密码由德军使用，后被盟军破译，成为加速战争结束的重要因素，由

此被载入史册。

Enigma 加密和解密需要一台类似打字机的设备,设定初始位置后,即可通过击打键盘输入明文,设备经过加密运作,输出(打印)为密文。解密也需要这一设备,通过巧妙的机械装置的切换,实现逆转(解密)操作,输入密文,输出即为明文。

如图 9.5 所示,Enigma 密码加密系统由一系列独立转动的圆柱体组成。电脉冲能够流经圆柱体,每个圆柱体有 26 个输入引脚与 26 个输出引脚,其内部连线将每个输入引脚连接到一个相应的输出引脚。

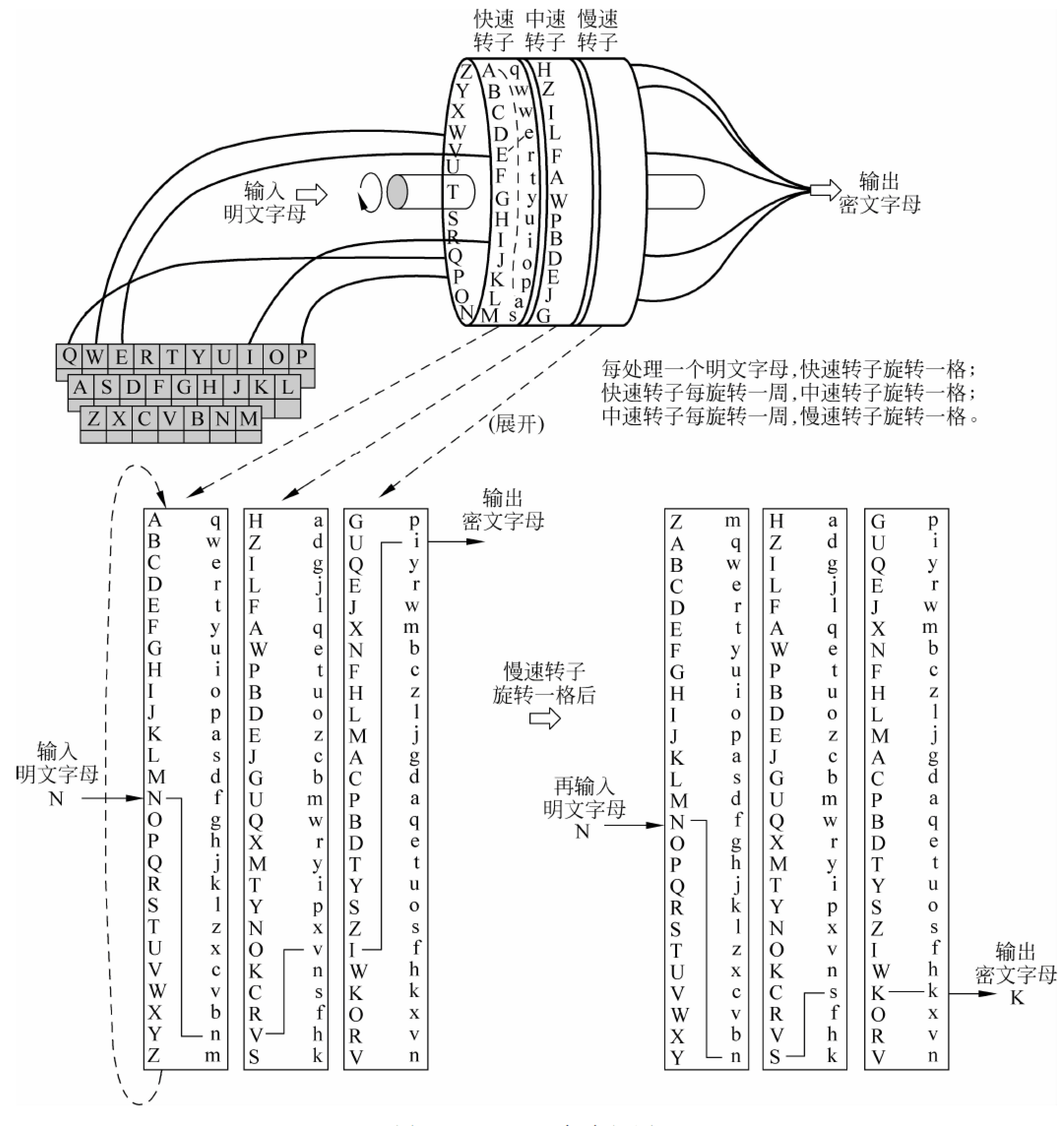


图 9.5 Enigma 加密机原理

如果每个输入引脚和输出引脚与字母表中的一个字母关联起来,则单个圆柱体定义了一个单字母替代。如果操作员按下某个字母键,一个电信号被加到第 i 个圆柱体的第 j 个引脚,并通过内部连接到达第 k 个输出引脚。

对于一个具有单圆柱体的机器，在每一个输入键被按下后，该圆柱体旋转一个位置，由此内部连接被相应移位，从而定义了一个不同的单字母替代密码。在明文的 26 个字母输入之后，圆柱体将回到原来的位置。这样，我们得到了周期为 26 的多字母替代算法。

当然一个单圆柱体的系统是不足以抵御令人生畏的密码攻击的。那么可以使用多个圆柱体，如采用快速、中速、慢速三个圆柱体。其中一个圆柱体的输出引脚与下一个圆柱体的输入引脚相连。快速圆柱体旋转一周，中速圆柱体旋转一个引脚位置；中速圆柱体旋转一周，慢速圆柱体旋转一个引脚位置。则总共可以获得 26×26×26＝17 576 种不同的替代字母可供使用。

Enigma 转子机密码是传统加密技术发展的顶峰。随着计算机的诞生，传统加密技术逐渐淡出人们的视线，取而代之的是现代加密技术。但 Enigma 密码的特殊地位仍然不可低估，不仅在于它具有超强的信息保密能力，更在于它指明了通向目前应用最为广泛的现代密码算法的技术途径。

CHAPTER 10

第10章 对称密钥加密

10.1 对称密钥加密原理

对称密钥加密(Symmetric Key Cryptography),又称为私钥加密(Private Key Cryptography)、单钥加密(One-Key Cryptography),因同一个密钥既用于加密又用于解密而得名。

作为一项重要的现代加密技术,对称密钥加密是信息保密的主要手段。其特点是:密钥(私钥)越长则加密强度越大。对称加密算法一般具有很高的计算效率,可面向大量数据,如文件、数据库、流媒体等的加密工作,但密钥的安全生成、安全分发、安全存储需要较大的管理工作量。

对称密钥加密依应用目标可分为流式加密和分组加密两大类。常用的对称密钥加密技术有Blowfish、TEA、DES、AES、IDEA、SMS4、RC4等,本章分别讨论其原理和详细算法。

10.2 流式加密

流式加密(Stream Cipher)技术一般面向以比特、字节为单位的信息流传输方式,通常要求连续、不间断地发送数据,而接收方同样持续接收并处理数据,加密和解密过程应与之紧密结合,既要不改变、不影响原有的传输方式,又要具备较高的运算效率和安全性。

RC4是Ron Rivest(也是RSA算法的发明人之一)于1987年设计的一种流式对称加密方法,采用可变密钥长度、面向字节操作。RC4算法执行效率高,密钥选取比较灵活,因此在SSL/TLS、802.11/WEP等许多网络应用系统中得到使用,成为最重要的流式加密算法。

RC4算法采用可变长密钥,密钥长度可在1～256B(8～2048b)之间随意选取,可对任意长度的明文实施加密和解密,密钥长度与被加密的信息长度无关。事实上可以做到加密一个字节发送一个字节、接收一个字节解密一个

字节，完全满足流媒体业务的需要。

RC4 算法以随机置换原理为基础，计算方法简便、高效。分析显示，当密钥长度达到 16B(128b)以上时，具有较强的抗攻击能力。

设密钥 K 的长度为 lB，$0<l\leqslant 256$，输入明文为 M，输出密文为 C。RC4 加密和解密算法分四步执行。

10.2.1　状态向量初始化

设：状态向量 $\boldsymbol{S}=\{S[0],\ S[1],\ S[2],\ \cdots,\ S[255]\}$，令 $S[i]=i$，$i=0,1,\cdots,255$，即状态相邻的 256 个元素依次赋值 0～255。

在整个计算过程中，$\boldsymbol{S}$ 的各个元素仅仅进行位置的交换(即数值换位)，而始终包含 0～255 这 256 个数值。

10.2.2　密钥初始化

设：临时向量 $\boldsymbol{T}=\{T[0],\ T[1],\ T[2],\ \cdots,\ T[255]\}$。将密钥 K 循环赋值给 $\boldsymbol{T}$，即 $T[i]=K[i \bmod l]$，$i=0,1,\cdots,255$。

此时，密钥 K 已经完成了使命，后续算法中不再使用。

10.2.3　初始置换

用向量 $\boldsymbol{T}$ 产生 $\boldsymbol{S}$ 的初始置换。根据 $T[i]$的值，将 $S[i]$与 $\boldsymbol{S}$ 中指定元素进行置换(即换位 swap)操作。置换方法用以下伪代码表示：

```
j = 0;
for  (i = 0  to  255)  do {
   j = ( j + S[ i ] + T[ i ] ) mod 256;
   swap( S[ i ], S[ j ] ); }
```

初始置换操作的本质就是根据密钥的值，改变原来 $\boldsymbol{S}$ 中呈升序排列的数值，使之成为比较随机的分布。至此，$\boldsymbol{T}$ 也不再使用，算法初始化阶段完成。

10.2.4　加密运算

对于输入明文 M，若长度为 nB，其字节流为 $M[k]$，$0\leqslant k<n$，经过加密运算，应输出密文 C，密文字节流为 $C[k]$，$0\leqslant k<n$。加密运算过程如下：

```
i = j = 0;
for  (k = 0  to  n)  do {
   i = ( i + 1 ) mod 256;
   j = ( j + S[ i ] ) mod 256;
   swap( S[ i ], S[ j ] );
   t = ( S[ i ] + S[ j ] ) mod 256;
   C[ k ] = M[ k ]⊕S[ t ];}
```

运算过程可以描述为：RC4 加密循环往复地使用状态向量 $\boldsymbol{S}$，每个明文字节与 $\boldsymbol{S}$ 生成的数值进行异或(⊕)操作，得到密文字节，同时 $\boldsymbol{S}$ 中的元素不断换位。长度值 n 实际上不起任何作用，加密过程可以不断进行下去。

RC4 解密的过程与加密完全一致,因为密钥相同,对 S 中的元素进行换位的机制相同,则解密方获得的"密钥流"是相同的,那么经过同一数值的异或操作,密文字节将还原为明文字节。

10.3 分组加密

10.3.1 分组加密原理

分组加密(Block Cipher)是以数据块为单位,进行整体变换,从明文块生成密文块,一块数据加密完成后继续加密下一块;解密运算也是按密文块依次进行。就加密算法本身而言,每一块数据的加密和解密都是相互独立的。

设分组大小为 nB。明文数据交付加密前,需要先进行数据填充(填充数据可为任意数值),如图 10.1 所示,使总长度为 nB 的整数倍。最后 4B 为长度字段,用以表示有效数据的长度。

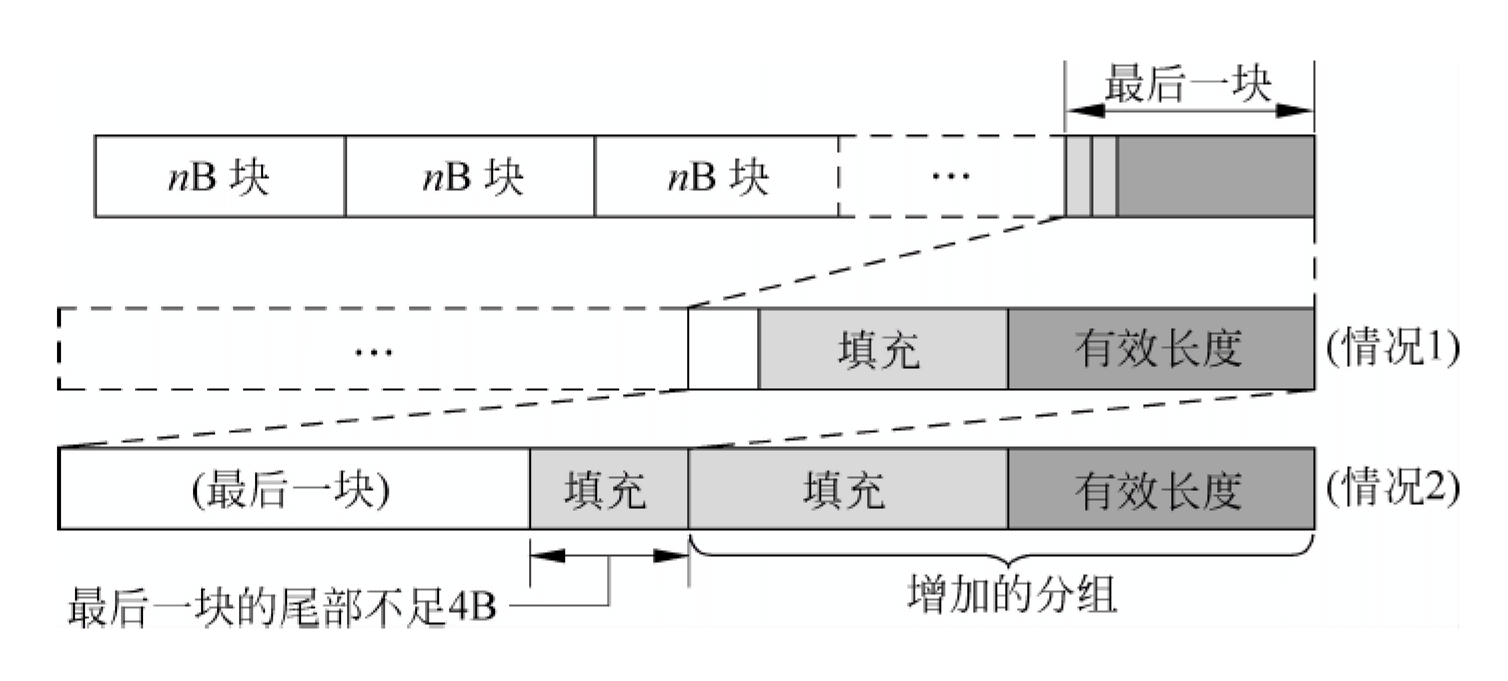


图 10.1 分组加密准备

思考:如果原始数据恰好是 nB 的整数倍应如何处理?

显然,对 nB 的分组,有 2^n 个不同的明文分组、2^n 个不同的密文分组,相互间共有 $2^n!$ 个非奇异的对应(变换)关系。

如图 10.2 所示,4b 明文分组 $M=(M_3M_2M_1M_0)$ 共有 $2^4=16$ 种编码组合,编码器的 16 个输出分别为 $X_0\sim X_f$。对于一种特定的对应转换关系,可一对一地映射为 $Y_0\sim Y_f$,进而可解码为 4b,输出 $C=(C_3C_2C_1C_0)$,C 即为明文 M 加密后的密文。例如,4b 明文为 1001(十六进制 9),编码结果应为 X_9,根据图 10.2 所示的对应关系,被映射为 Y_7,则解码为密文 0111。

这一分组编码、解码及变换过程就是分组加密的基本原理,且过程是可逆的,即分组解密遵循同样的原理。

如果改变图 10.2 中的变换关系,对同一个明文会输出不同的密文,相当于采用了不同的密钥,产生不同的加密结果。换言之,变换关系本身就是密钥,每一种变换就是一个密钥。

由于这个加密模型穷尽了所有的编码和映射关系,因此是理想分组加密方法,就是说无法做得比这更好了。然而,该模型存在一个不容忽视的问题:密钥过长。为了证明这一点,不妨将 4b 明文的所有编码按升序排列,将密文与明文对应列出。以图 10.2 所示的变换,可得如图 10.3 所示的映射关系表。每一种不同变换均可产生一张唯一的映射关系表(其中明文编码统一为升序排列,都是相同的),总共有 $2^n!$ 张表。

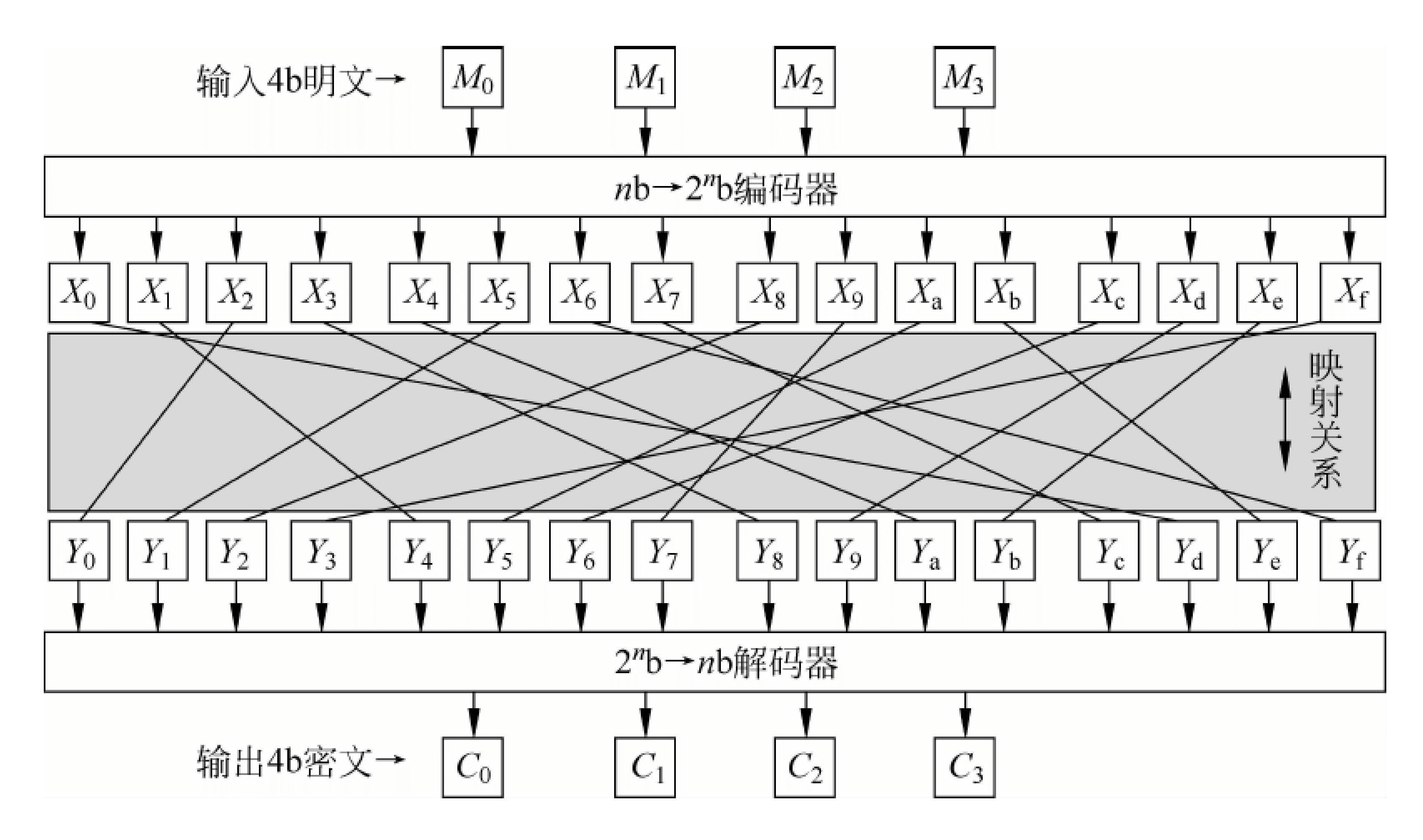


图 10.2　4b 分组加密变换原理

明文	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
密文	1101	0100	0000	1000	1010	0001	1111	1100	0010	0111	0101	1110	0110	1001	1011	0011

图 10.3　4b 分组加密映射关系

因此，既然变换关系就是密钥，则 4b 分组加密系统的每一种变换关系就需要一个 $4\times 2^4=64$b 的密钥。那么，nb 分组加密系统的密钥长度就是 $n\times 2^n$b。对于常用的 64b 分组，其密钥长度将达到惊人的 10^{21}b。过长的密钥对数据加密的计算、管理均会带来不利影响。

为此，分组加密算法设计实际上采用的是**理想分组加密的近似体制**，取不同映射关系的一个子集，以减少映射关系数为代价，换取密钥长度的缩短，使加密方法具有实用性。

仍然以 4b 分组加密为例。设明文为(p_1,p_2,p_3,p_4)，密文为(C_1,C_2,C_3,C_4)。定义如下线性方程组(类似 Hill 密码)：

$$\begin{cases} C_1=(k_{11}p_1+k_{12}p_2+k_{13}p_3+k_{14}p_4) \\ C_2=(k_{21}p_1+k_{22}p_2+k_{23}p_3+k_{24}p_4) \\ C_3=(k_{31}p_1+k_{32}p_2+k_{33}p_3+k_{34}p_4) \\ C_4=(k_{41}p_1+k_{42}p_2+k_{43}p_3+k_{44}p_4) \end{cases}$$

或以矩阵表示为

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix}=\begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{pmatrix}\begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}$$

其中 k_{ij}，$i=1,2,3,4$，$j=1,2,3,4$ 为一位二进制比特。如果将$(k_{11},k_{12},\cdots,k_{44})$作为密钥，则 4b 分组加密算法成立，且支持逆变换，可以解密。此例中，密钥长度缩短为 $4^2=16$b。

同理推广到 nb 分组加密，密钥长度将从理想分组加密方法的 $n\times 2^n$b 大大缩短为 n^2b。本例说明密钥长度缩短是可行的，当然保密性能也会相应降低。如果加密算法设计合理，就会在密钥长度和加密能力间找到适当的平衡点。

10.3.2 Feistel 加密模型

根据信息论创始人香农(Claude Shannon)的加密理论,使用信息编码的混淆(confusion)和扩散(diffusion)方法,可以抵抗基于统计方法的密码分析(破解)。混淆可以尽可能使密文和密钥间的统计关系复杂化,阻止发现密钥;扩散则使明文的统计特征消散在密文中,可以让每个明文编码单元尽可能多地影响多个密文编码单元。考察下式:

$$c_n = \sum_{i=1}^{k} m_{n+i} (\bmod 26)$$

式中:m 为明文字母,c_n 为密文字母,则每个 m_j 都会影响到 k 个密文字母,就是一种扩散方法的例子。

在对称密钥加密运算中,常用的代换(substitution)和置换(swap)分别就是混淆和扩散的实际应用。代换采用函数运算方法,置换通过数据的交叉换位实现。**Feistel 加密模型**就是基于这一理论设计的。

设:明文和密文为 $2wb$,密钥为 K。图 10.4 所示为 Feistel 加密的算法结构。该模型由 i 轮加密运算组成,每轮具有相似的运算方法,分别使用由密钥 K 生成的该轮次的子密钥 K_i。每轮的输入被分为 wb 的左半部分和右半部分,进行代换运算,在输出前左、右部分进行置换运算。算法结束前再进行一次左右置换运算,最终合并成 $2wb$ 的密文。

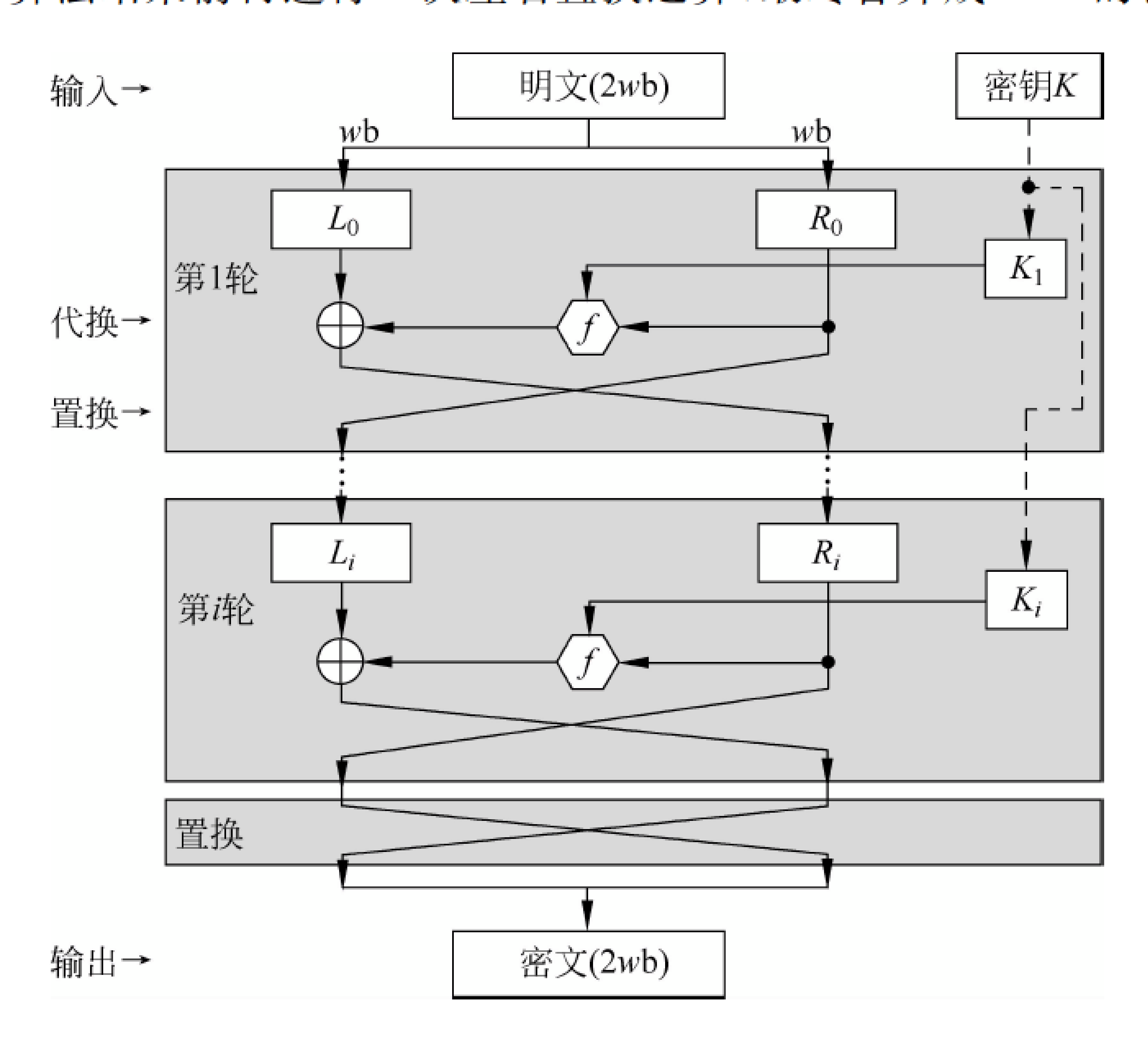


图 10.4 Feistel 加密模型算法结构

解密过程与加密过程完全一致(而不是逆向进行),但应逆序使用子密钥。证明如下。

不失一般性,设加密算法执行 16 轮。并设:加密时,明文为 $LE_0 | RE_0$,输出密文为 $RE_{16} | LE_{16}$;解密时,密文为 $LD_0 | RD_0$,输出密文为 $RD_{16} | LD_{16}$。

显然有:$LD_0 = RE_{16}$;$RD_0 = LE_{16}$。

对于加密过程:$LE_{16} = RE_{15}$;$RE_{16} = LE_{15} \oplus f(RE_{15}, K_{16})$。

对于解密过程第 1 轮:$LD_0 = RE_{16}$;$RD_0 = LE_{16}$;$\mathbf{LD_1} = RD_0 = LE_{16} = \mathbf{RE_{15}}$;

$$\boldsymbol{RD_1} = LD_0 \oplus f(RD_0, K_{16}) = RE_{16} \oplus f(RE_{15}, K_{16})$$
$$= [LE_{15} \oplus f(RE_{15}, K_{16})] \oplus f(RE_{15}, K_{16}) = \boldsymbol{LE_{15}}。$$

因此，解密第 1 轮输出 $\boldsymbol{LE_{15}} | \boldsymbol{RE_{15}}$，正是加密第 16 轮输入左右互换的值。

依此类推，解密第 16 轮：$LD_{16} = RE_0$；$RD_{16} = LE_0$；最终换位输出为 $\boldsymbol{LE_0} | \boldsymbol{RE_0}$，正是原始明文。证毕。

Feistel 加密模型是对称密钥加密算法的指导性技术框架，TEA、Blowfish、DES 等算法均符合这一模型。加密过程采用的 i 轮次运算又称为迭代(iteration)操作。迭代轮数越多，密文信息的混淆和扩散越彻底，保密性就越强，但会带来加解密速度变慢的问题，因此通常选择 16 轮迭代。此外，分组长度和密钥长度越长，安全性能也越好，但同样存在拖慢运算速度的问题，常见的分组长度选择 64b 或 128b，密钥长度则会根据算法指标、密级要求来确定。

10.3.3　TEA 算法

微型加密算法(Tiny Encryption Algorithm，TEA)是一种轻量级的对称密钥分组加密算法，由剑桥大学的 David Wheeler 和 Roger Needam 提出。算法符合 Feistel 加密模型。

TEA 不是通过算法的复杂性来保证安全，而是依赖加密迭代的轮次数(可选 16、32 或 64 轮)。算法执行速度快，简单易实现，并具有较强的抗差分攻击能力，适合信息保密强度要求不高，但通信效率要求较高的应用，如即时通信系统。

TEA 算法使用了一个神秘常数 δ，以保证每一轮加密有所差异。而 δ 来源于黄金比率 φ：

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.618\,033\,988\,7\cdots$$

定义 δ：

$$\delta = \frac{2^{32}}{\varphi} = (\sqrt{5}-1) \times 2^{31} = 0x9e3779b9$$

TEA 面向 64b 分组明文进行加密，密钥长度为 128b。TEA 加密和解密算法(32 轮迭代)流程如下：

```
/* 输入明文为 t[2],密钥为 k[4],数组元素均为 32b */
void TEA_encrypt (uint32 *t, uint32 *k)
{
   uint32 delta = 0x9e3779b9, s = 0;

   for (i = 0; i < 32; i++)
     {
        s += delta;
        t[0] += ((t[1]<<4) + k[0]) ⊕ (t[1] + s) ⊕ ((t[1]>>5) + k[1]);
        t[1] += ((t[0]<<4) + k[2]) ⊕ (t[0] + s) ⊕ ((t[0]>>5) + k[3]);
   }
}                              /* 函数子程序返回后,t[0]|t[1]为密文 */

/* 输入密文为 t[2],密钥为 k[4],数组元素均为 32b */
void TEA_decrypt (uint32 *t, uint32 *k)
```

```
{
  uint32 delta = 0x9e3779b9, s = 0xc6ef3720;

    for (i = 0; i < 32; i++)
      {
        t[1] -= ((t[0]<<4) + k[2] ⊕ (t[0] + s) ⊕ ((t[0]>>5) + k[3]);
        t[0] -= ((t[1]<<4) + k[0] ⊕ (t[1] + s) ⊕ ((t[1]>>5) + k[1]);
        s -= delta;
      }
}                        /* 函数子程序返回后,t[0]|t[1]为明文 */
```

算法流程中,加法和减法运算均作 mod 2^{32},$\oplus$为按位异或运算,"<<"和">>"分别表示 32b 逻辑左移和逻辑右移运算。

如图 10.5 所示,TEA 算法的一个循环包含了两轮相似的迭代运算,不同之处仅为使用了不同的子密钥。程序代码在实现代换运算的同时,巧妙地实现了隐含的换位置换操作。

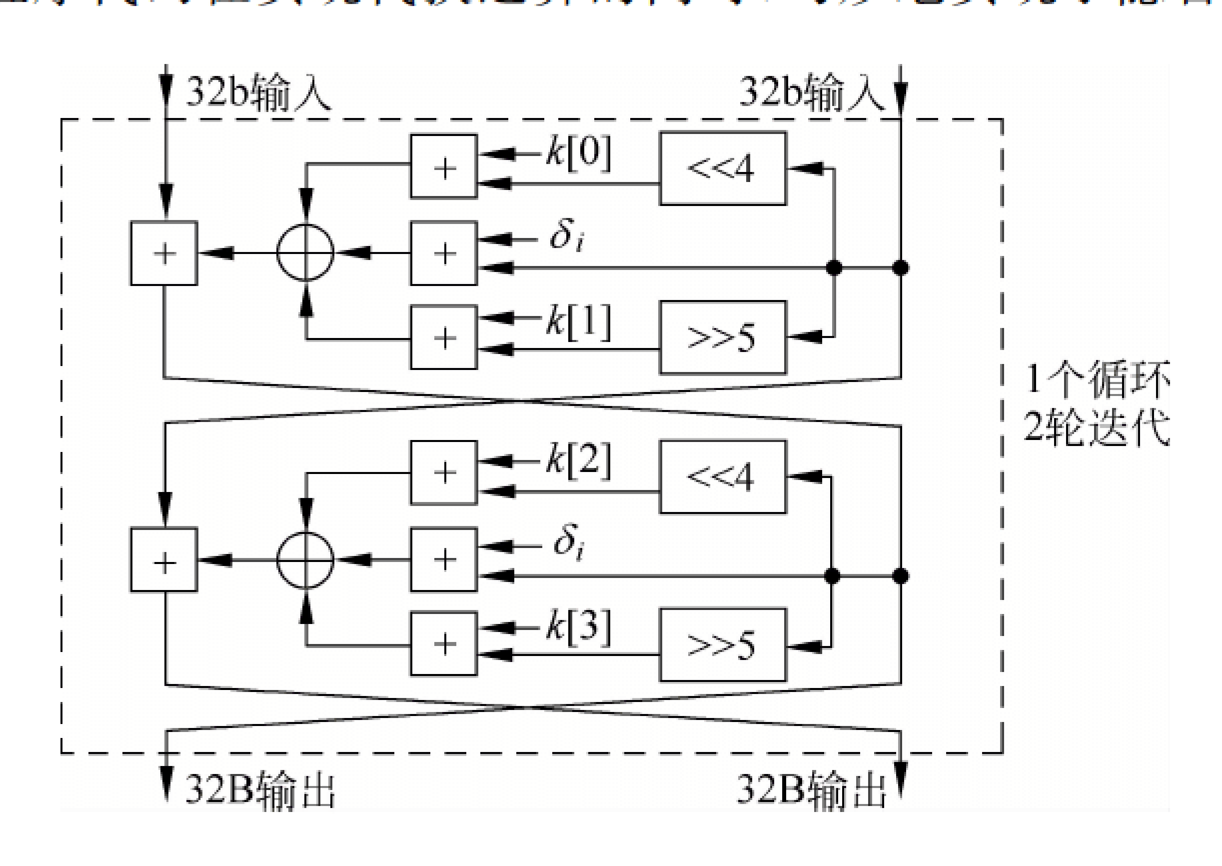


图 10.5　TEA 算法加密运算结构

10.3.4　Blowfish 算法

Blowfish 算法是 1993 年由布鲁斯·施奈尔(Bruce Schneier)开发的对称密钥分组加密算法,符合 Feistel 加密模型,分组长度为 64b,密钥长度从 1 至 448b 可变。与 DES 等算法相比,Blowfish 算法具有非常紧凑、易于实现、处理速度很快、无须授权即可使用等特点,广泛用于 SSH、文件加密软件等应用。

Blowfish 是一种会鼓气的鱼,遇到危险时会把身体鼓成大大的圆球状,完全隐藏其本来面目,并用庞大的体形来吓退强敌。

Blowfish 算法分两个部分:密钥准备(密钥扩展)和数据加密。

1. 密钥准备

Blowfish 算法使用大量的子密钥(subkey)。在加密或解密过程开始前,应先将密钥 K 转换为各迭代轮次所需的子密钥,总长度为 4168B。

定义 P 数组为 *p-array*[1..18],S 盒为二维数组 *s-box*[1..4][0..255],数组元素均为 32b 类型,用于存放子密钥。

Blowfish 子密钥的生成分为 4 步,其中需要使用 Blowfish 加密算法(见步骤(2);但不

用担心，不会发生“先有鸡还是先有蛋”的两难问题）。

(1) 用固定的值初始化 *p-array* 和 *s-box* 数组：

p-array[1] = 0x243f6a88，*p-array*[2] = 0x85a308d3，*p-array*[3] = 0x13198a2eh，*p-array*[4]=0x03707344，*p-array*[5]=0xa4093822，…，*p-array*[18]=0x8979fb1b；*s-box*[1][0]=0xd1310ba6，*s-box*[1][1]=0x98dfb5ac，…，*s-box*[1][255]=0x6e85076a，*s-box*[2][0]=0x4b7a70e9，*s-box*[2][1]=0xb5b32944，*s-box*[2][2]=0xdb75092e，…

(2) 循环使用 K，顺序取 K 的 32b 数值，依次对 *p-array*[1]到 *p-array*[18]作异或运算（相当于把 K 变成 18×32b 的等效密钥 $KK\cdots K$）。设 64b 变量 $Z=0$。

(3) 使用到目前为止得到的 *p-array* 和 *s-box*，对 Z 进行 Blowfish 加密运算，得到更新后的 Z。

(4) 赋值 *p-array*[1] = *p-array*[2]=Z，循环返回步骤(3)。当下一次循环到这一步时，应赋值 *p-array*[3]和 *p-array*[4]，依次类推，直到所有 *p-array* 元素更新完毕，接着更新所有 *s-box* 元素。

子密钥生成循环共需进行 521 轮，之后，子密钥已经在 *p-array* 和 *s-box* 中准备好，可以进行正式的数据加密运算了。

2. 数据加密

Blowfish 算法数据加密过程采用 16 轮 Feistel 迭代结构。设输入 64b 明文为 X。将 X 拆分为左、右两部分，成为两个 32b 数据 X_L 和 X_R。

算法采用的 f 函数是对 X_L 进行操作。X_L 被均分为 4 个 8b 数据 a，b，c 和 d。函数表达式如下(式中加法运算均需 mod 2^{32}，$\oplus$为异或运算)：

$$f(X_L) = ((s\text{-}box[1][a] + s\text{-}box[2][b]) \oplus s\text{-}box[3][c]) + s\text{-}box[4][d]$$

加密运算如下：

```
For i  =  1 to 16 Do            /* 总共 16 轮迭代运算 */
  XL =  XL⊕p-array[i]
  XR =  f (XL)⊕XR
  Swap(XL, XR)                  /* 每轮结束前的换位置换操作 */
Next i
Swap(XL, XR)                    /* 算法结束前的换位置换操作 */
XR =  XR⊕p-array[17]
XL =  XL⊕p-array[18]
```

最后，重新拼接 X_L 和 X_R 即为 64b 密文，加密完成。解密过程与加密完全相同，只是逆序使用 *p-array*[1..18]。

10.3.5　SMS4 算法

SMS4 算法是基于 Feistel 模型的对称密钥分组加密算法，作为中国的第一个商用密码标准，于 2006 年 1 月发布，在 WiFi 的中国技术标准 WAPI 等体系中得到应用。

SMS4 采用的数据分组长度和密钥长度均为 128b，数据处理单位为 8b 和 32b，进行 32 轮非线性迭代运算。

如图 10.6 所示，SMS4 将 128b 的明文分割为 4 个 32b 分组块，采用 8 次循环，每次使

用轮函数 f 进行 4 轮迭代,每轮使用一个子密钥(轮密钥)。子密钥由 128b 的密钥经扩展函数 g 生成。最后,反向排列 32b 的 $X_{32} \sim X_{35}$,合并为 128b 的密文。

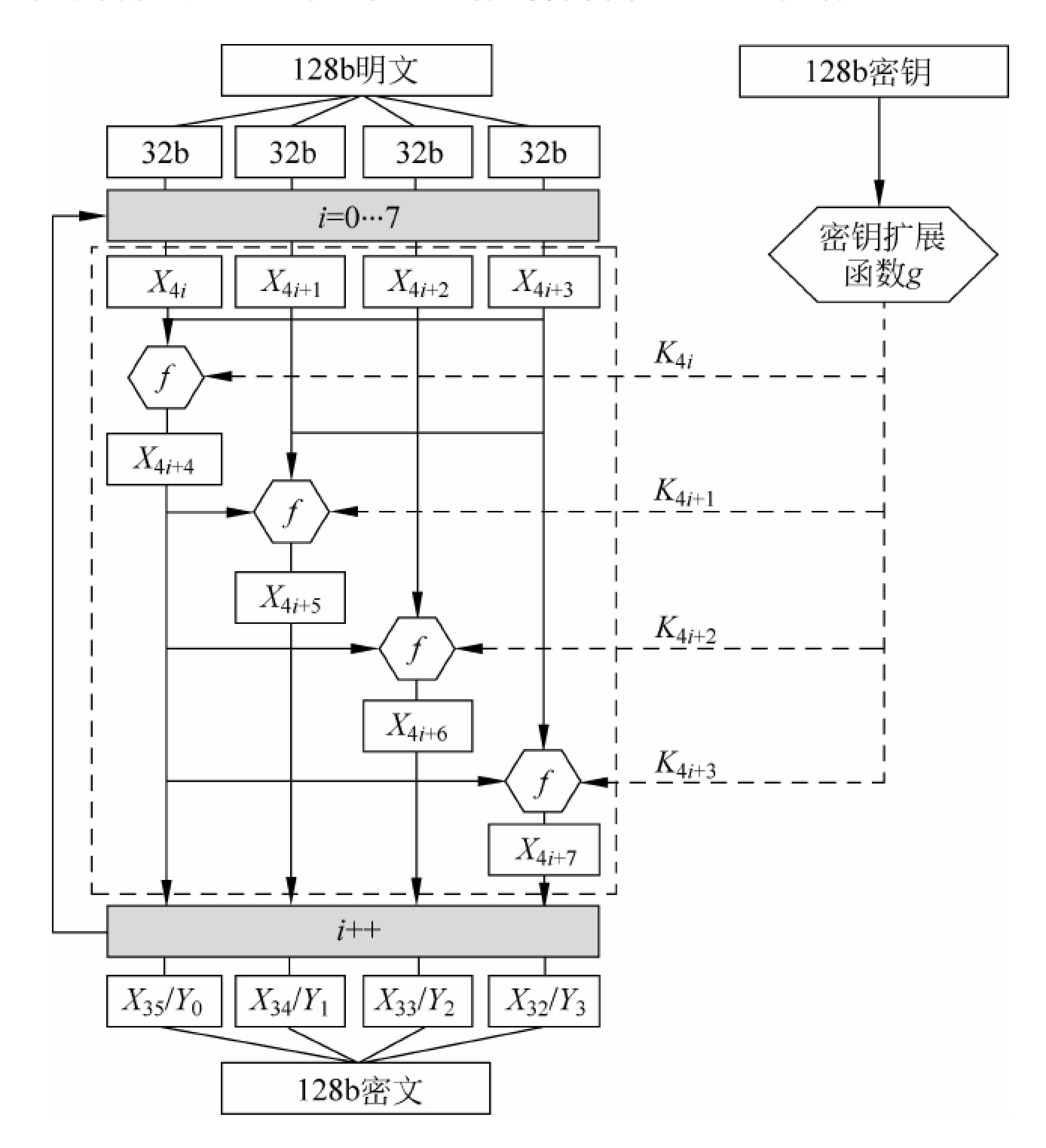


图 10.6　SMS4 加密算法结构

SMS4 的加密运算过程可用如图 10.7 所示的等价变换图表示。

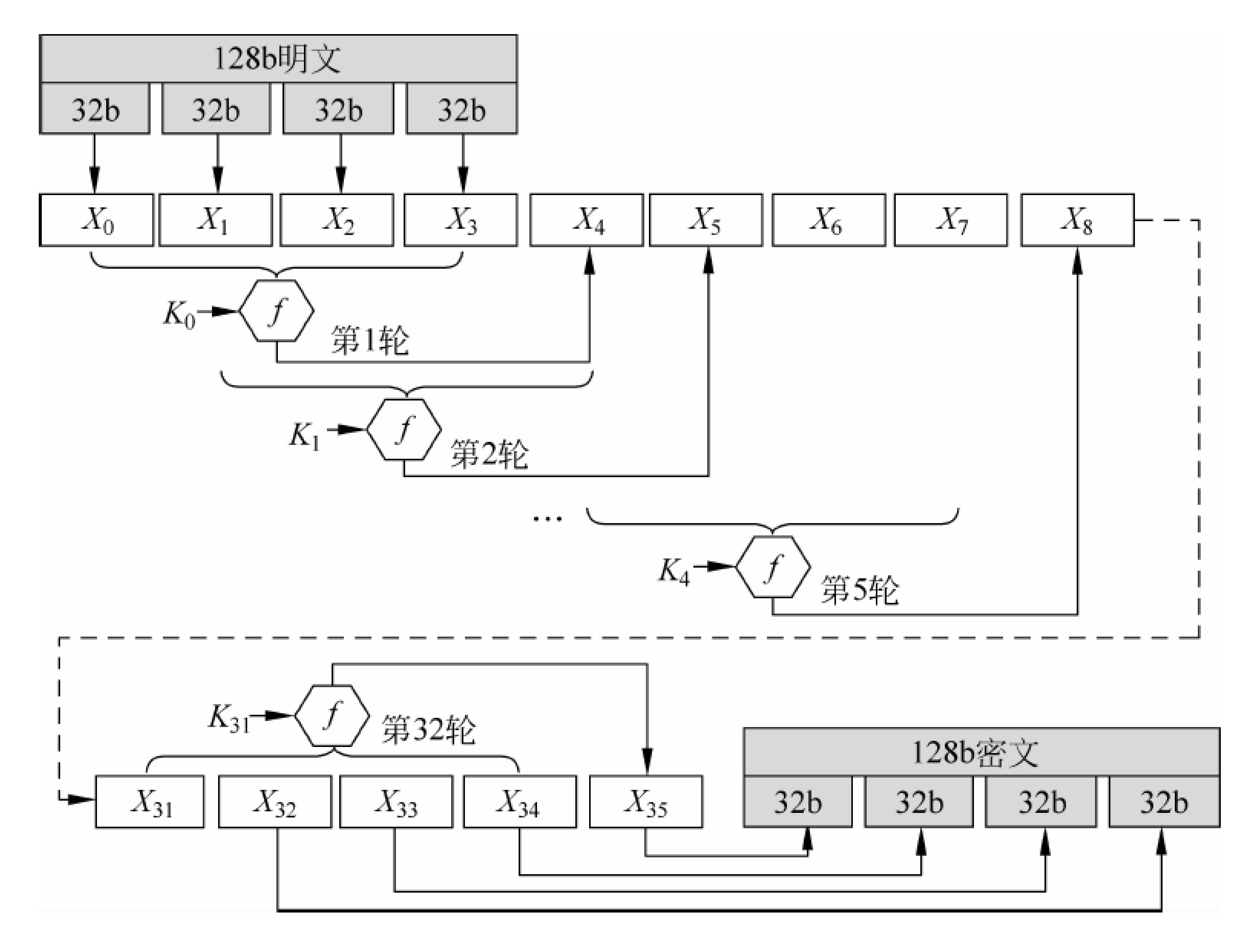


图 10.7　SMS4 加密算法等价变换

设 X_j 为 32b 寄存器，k_i 为轮密钥。SMS4 每轮的迭代运算由按位异或(⊕)、循环左移(<<<)和变换函数 S 所组成，如图 10.8 所示，轮函数 f 定义如下：

$$X_{i+4} = f(X_i, X_{i+1}, X_{i+2}, X_{i+3}, k_i)$$
$$= X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus k_i),\quad i = 0,1,\cdots,31$$
$$T(x) = L(S(x))$$
$$L(x) = x \oplus (x <<< 2) \oplus (x <<< 10) \oplus (x <<< 18) \oplus (x <<< 24)$$

其中 32b 的 $S(x)$ 函数由 4 个并列的 8b 单元的 S-box 变换所构成，算法如图 10.9 所示。

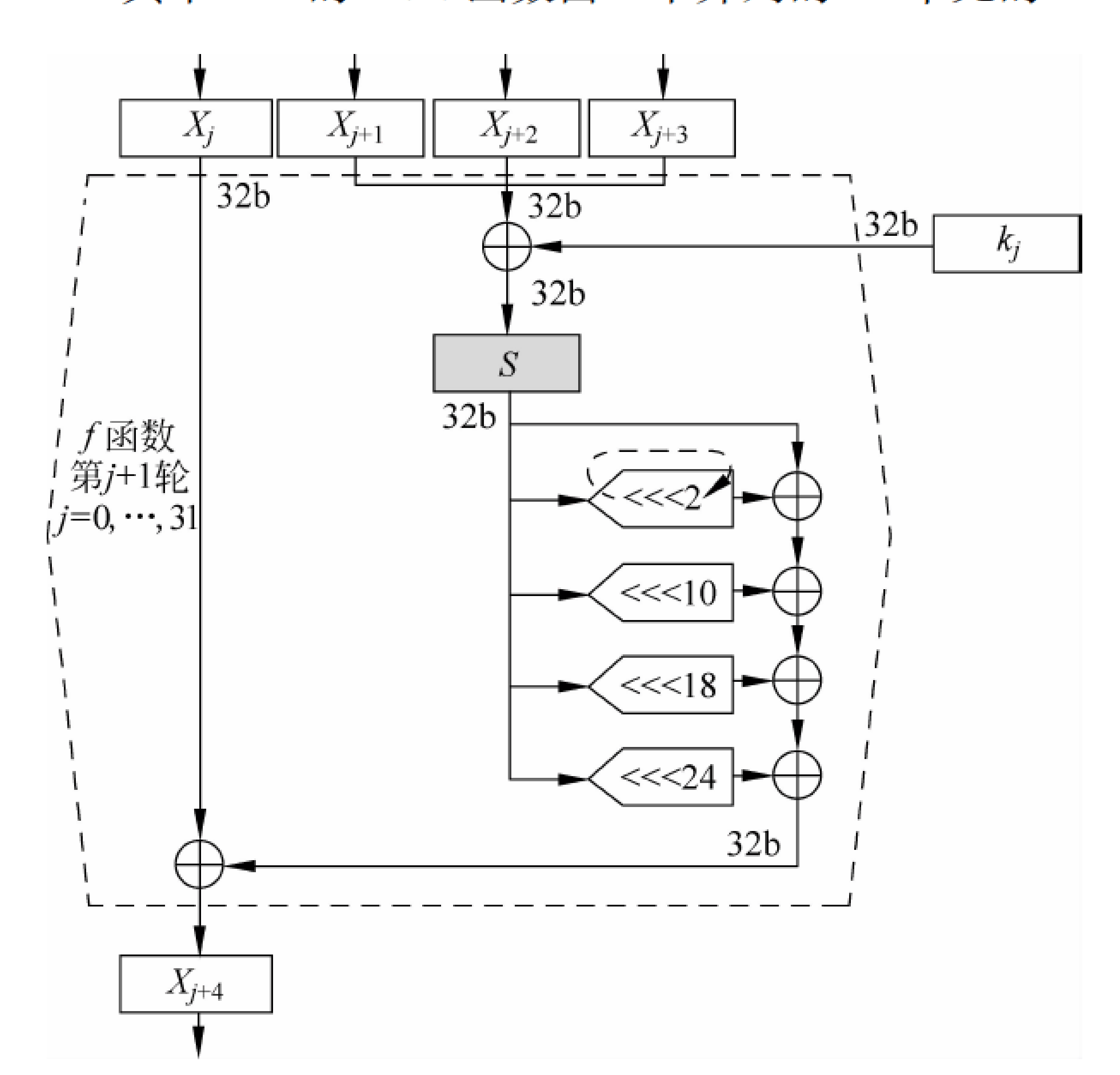


图 10.8　SMS4 加密算法轮函数

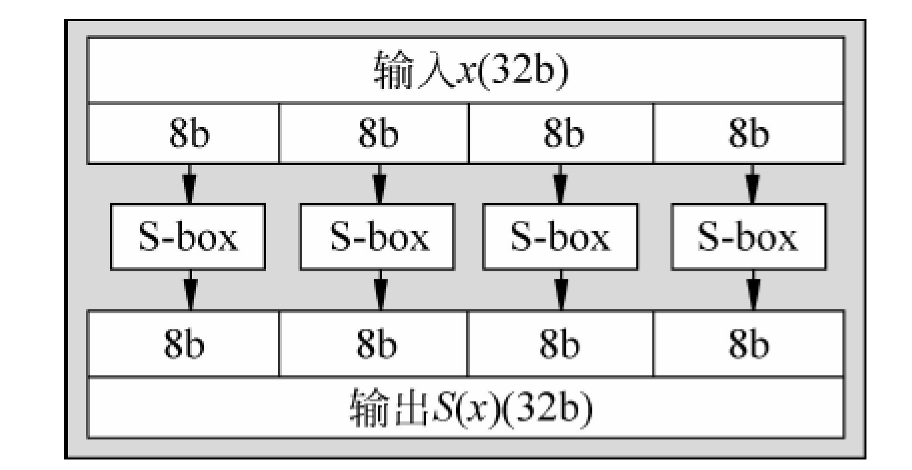


图 10.9　SMS4 算法 S 函数

S-box 采用查表选择法，使用如表 10.1 所示的置换选择表，属于一种非线性变换方式。表中的 256 个数值从 0x00 到 0xff，具有唯一性。设 S-box 的 8b 输入为十六进制 0xRC，S-box 以 0x0R 为行号、0x0C 为列号进行查表操作，查表所得的数据作为 S-box 的输出结果。例如，0x2a 经 S-box 输出为 0x0b。

表 10.1　S-box 置换选择表

R \ C	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	d6	90	e9	fe	cc	e1	3d	b7	16	b6	14	c2	28	fb	2c	05
1	2b	67	9a	76	2a	be	04	c3	aa	44	13	26	49	86	06	99
2	9c	42	50	f4	91	ef	98	7a	33	54	0b	43	ed	cf	ac	62
3	e4	b3	1c	a9	c9	08	e8	95	80	df	94	fa	75	8f	3f	a6
4	47	07	a7	fc	f3	73	17	ba	83	59	3c	19	e6	85	4f	a8
5	68	6b	81	b2	71	64	da	8b	f8	eb	0f	4b	70	56	9d	35
6	1e	24	0e	5e	63	58	d1	a2	25	22	7c	3b	01	21	78	87
7	d4	00	46	57	9f	d3	27	52	4c	36	02	e7	a0	c4	c8	9e
8	ea	bf	8a	d2	40	c7	38	b5	a3	f7	f2	ce	f9	61	15	a1

续表

R \ C	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
9	e0	ae	5d	a4	9b	34	1a	55	ad	93	32	30	f5	8c	b1	e3
a	1d	f6	e2	2e	82	66	ca	60	c0	29	23	ab	0d	53	4e	6f
b	d5	db	37	45	de	fd	8e	2f	03	ff	6a	72	6d	6c	5b	51
c	8d	1b	af	92	bb	dd	bc	7f	11	d9	5c	41	1f	10	5a	d8
d	0a	c1	31	88	a5	cd	7b	bd	2d	74	d0	12	b8	e5	b4	b0
e	89	69	97	4a	0c	96	77	7e	65	b9	f1	09	c5	6e	c6	84
f	18	f0	7d	ec	3a	dc	4d	20	79	ee	5f	3e	d7	cb	39	48

SMS4 加密算法每轮使用不同的 32b 子密钥，每个子密钥都是由 128b 原始密钥 K 经以下算法生成。

设 4 个 32b 常数 FK_i，$i=0,1,2,3$，$FK_0=0\text{xA3B1BAC6}$，$FK_1=0\text{x56AA3350}$，$FK_2=0\text{x677D9197}$，$FK_3=0\text{xB27022DC}$。

又设 32b 参数 CK_i，$i=0,1,\cdots,31$，定义

$$CK_i = < ck[i][0] \mid ck[i][1] \mid ck[i][2] \mid ck[i][3] >$$

即 CK_i 是由 4 个一组的 8b 数值 $ck[i][j]$，$j=0,1,2,3$ 拼合而成，而 $ck[i][j]$可根据其下标进行取值：

$$ck[i][j] = (4i+j)\times 7 \pmod{256}$$

由此可得 32 个 CK_i 参数为

00070e15，1c232a31，383f464d，545b6269，
70777e85，8c939aa1，a8afb6bd，c4cbd2d9，
e0e7eef5，fc030a11，181f262d，343b4249，
50575e65，6c737a81，888f969d，a4abb2b9，
c0c7ced5，dce3eaf1，f8ff060d，141b2229，
30373e45，4c535a61，686f767d，848b9299，
a0a7aeb5，bcc3cad1，d8dfe6ed，f4fb0209，
10171e25，2c333a41，484f565d，646b7279。

将密钥 K 分割为 4 个 32b 数值，即 $K=(K_0,K_1,K_2,K_3)$。设每轮使用的子密钥为 k_i，$i=0,1,\cdots,31$，Z_j 为中间变量，密钥变换算法如下：

$$Z_j = K_j \oplus FK_j,\quad j=0,1,2,3$$

$$k_i = Z_{i+4} = Z_i \oplus T'(Z_{i+1} \oplus Z_{i+2} \oplus Z_{i+3} \oplus CK_i)$$

$$T'(x) = L'(S(x))$$

$$L'(x) = x \oplus (x \ll 13) \oplus (x \ll 23)$$

可见，SMS4 子密钥的生成同样采用了加密运算中的 S 函数变换，迭代运算方法也很类似，便于算法的实现。

SMS4 算法的解密过程与加密完全相同，但逆序使用子密钥。

10.3.6 DES 算法

数据加密标准(Data Encryption Standard，DES)是信息加密领域最著名、应用最广泛的

技术，于 1977 年作为美国联邦标准发布，采用 DEA(Data Encryption Algorithm)算法，但习惯上就以 DES 指代 DEA。

DES 针对 64b 数据块(分组)进行加密和解密操作，属于对称密钥分组加密算法。DES 采用的密钥被称为会话密钥(Session Key)，可以是 64b 的任意数。密钥每字节的最高位作为奇偶校验位，因此实际密钥长度为 56b。

任何输入明文被划分成 64b 分组单元，DES 对每个分组进行 16 轮迭代运算，每轮使用一个 48b 的子密钥，而子密钥由 56b 的原始密钥变换而来。DES 加密算法如下。

首先，对 64b(编号为 bit_1～bit_{64})输入分组进行初始置换(Initial Permutation，IP)，并把输出分为 32b 的 L_0、R_0 两部分。置换规则如表 10.2 所示。

表 10.2　DES 初始置换表

L_0(32b)	R_0(32b)
58，50，12，34，26，18，10，2，60，52，44，36，28，20，12，4， 62，54，46，38，30，22，14，6，64，56，48，40，32，24，16，8，	57，49，41，33，25，17，9，1，59，51，43，35，27，19，11，3， 61，53，45，37，29，21，13，5，63，55，47，39，31，23，15，7

然后，DES 进入 16 轮的迭代运算 f，包括扩展换位运算、选择压缩运算、置换运算等步骤，每轮使用一个子密钥 K_i。运算过程如表 10.3 所示。

表 10.3　DES 迭代运算 f

	L_{i-1}(32b)	R_{i-1}(32b)
16 轮 迭代运算 f $i=1,\cdots,16$	↓	扩展换位运算 E → 48b
		与 K_{i-1} 按位异或运算(48b)
		选择压缩运算 S → 32b
		置换运算 P → 32b
		与 L_{i-1} 按位异或运算(32b)
	L_i(32b)← R_{i-1}	R_i(32b)

经过 16 次迭代运算 f 后，得到 L_{16} 和 R_{16}，以此作为输入，进行如表 10.4 所示的逆置换(IP^{-1})操作，即可得到输出的 64b 密文。IP^{-1} 正好是 IP 的逆变换，例如，bit_1 经过 IP 后，在 bit_{40} 位置；而通过 IP^{-1} 后，bit_{40} 换回到 bit_1。

表 10.4　DES 逆置换表

$L_{16}+R_{16}$(64b)	
40，8，48，16，56，24，64，32，	39，7，47，15，55，23，63，31，
38，6，46，14，54，22，62，30，	37，5，45，13，53，21，61，29，
36，4，44，12，52，20，60，28，	35，3，43，11，51，19，59，27，
34，2，42，10，50，18，58，26，	33，1，41，9，49，17，57，25

扩展换位运算 E 实现 32b 到 48b 变换，如表 10.5 所示。

表 10.5　DES 扩展换位运算 E

R_i(32b)→48b															
32	1	2	3	4	5	4	5	6	7	8	9	8	9	10	11
12	13	12	13	14	15	16	17	16	17	18	19	20	21	20	21
22	23	24	25	24	25	26	27	28	29	28	29	30	31	32	1

置换运算 P 实现 32b 的换位操作,如表 10.6 所示。

选择压缩运算 S 如图 10.10 所示。输入的 48b 数据被均分为 8 个,各为 6b,分别由 S 所包含的 S_1～S_8 八个选择函数(S-box)进行处理。每个选择函数的功能是把 6b 数据经变换得到 4b 数据,最终把 48b 输入值压缩转换为 32b。

表 10.6　DES 置换运算 P

32b															
16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

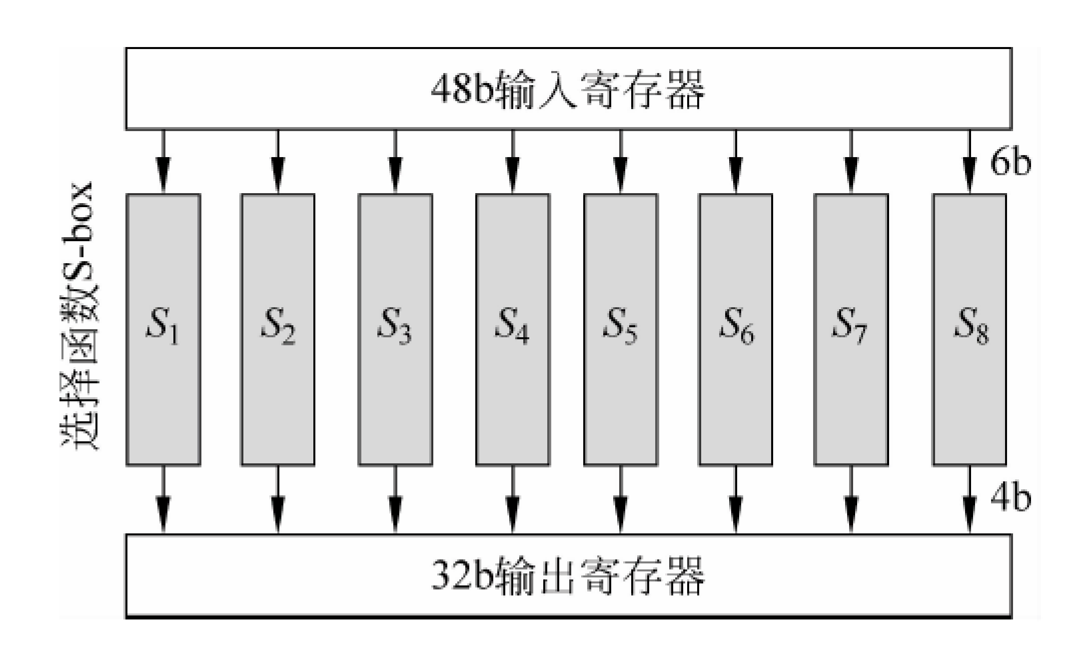


图 10.10　DES 选择压缩运算 S

选择函数 S_1～S_8 采用查表变换方法。如表 10.7 所示,每个 S-box 均占有 4 行(0～3 行)、每行 16 列(0～15 列)的数值矩阵。设输入 6b 数据为 $D_1D_2D_3D_4D_5D_6$,令列号 $=D_2D_3D_4D_5$(取值范围为 0,…,15),行号 $=D_1D_6$(取值范围为 0,…,3),查表得到对应的数,以 4b 二进制表示,即得一个 S-box 的输出。例如,S_2 输入 110110,则行、列号为(2,11),查表得 6,输出为 0110。

表 10.7　DES 选择压缩运算的 S-box 表

S_1: 14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7, 0,15,7,4,14,2,13,1,10,6,12,11,9,5,3,8, 4,1,14,8,13,6,2,11,15,12,9,7,3,10,5,0, 15,12,8,2,4,9,1,7,5,11,3,14,10,0,6,13	S_5: 2,12,4,1,7,10,11,6,8,5,3,15,13,0,14,9, 14,11,2,12,4,7,13,1,5,0,15,10,3,9,8,6, 4,2,1,11,10,13,7,8,15,9,12,5,6,3,0,14, 11,8,12,7,1,14,2,13,6,15,0,9,10,4,5,3
S_2: 15,1,8,14,6,11,3,4,9,7,2,13,12,0,5,10, 3,13,4,7,15,2,8,14,12,0,1,10,6,9,11,5, 0,14,7,11,10,4,13,1,5,8,12,6,9,3,2,15, 13,8,10,1,3,15,4,2,11,6,7,12,0,5,14,9	S_6: 12,1,10,15,9,2,6,8,0,13,3,4,14,7,5,11, 10,15,4,2,7,12,9,5,6,1,13,14,0,11,3,8, 9,14,15,5,2,8,12,3,7,0,4,10,1,13,11,6, 4,3,2,12,9,5,15,10,11,14,1,7,6,0,8,13

续表

S_3： 10,0,9,14,6,3,15,5,1,13,12,7,11,4,2,8, 13,7,0,9,3,4,6,10,2,8,5,14,12,11,15,1, 13,6,4,9,8,15,3,0,11,1,2,12,5,10,14,7, 1,10,13,0,6,9,8,7,4,15,14,3,11,5,2,12	S_7： 4,11,2,14,15,0,8,13,3,12,9,7,5,10,6,1, 13,0,11,7,4,9,1,10,14,3,5,12,2,15,8,6, 1,4,11,13,12,3,7,14,10,15,6,8,0,5,9,2, 6,11,13,8,1,4,10,7,9,5,0,15,14,2,3,12
S_4： 7,13,14,3,0,6,9,10,1,2,8,5,11,12,4,15, 13,8,11,5,6,15,0,3,4,7,2,12,1,10,14,9, 10,6,9,0,12,11,7,13,15,1,3,14,5,2,8,4, 3,15,0,6,10,1,13,8,9,4,5,11,12,7,2,14	S_8： 13,2,8,4,6,15,11,1,10,9,3,14,5,0,12,7, 1,15,13,8,10,3,7,4,12,5,6,11,0,14,9,2, 7,11,4,1,9,12,14,2,0,6,10,13,15,3,5,8, 2,1,14,7,4,10,8,13,15,12,9,0,3,5,6,11

16 轮迭代运算 f 中，每轮使用的子密钥 $K_0 \sim K_{15}$ 来自 64b 初始会话密钥 K($\text{bit}_0 \sim \text{bit}_{63}$)。先实施密钥缩小选择换位 1(如表 10.8 所示)，同时也去除了校验位(每字节的最高位，如 bit_7)，得到各 28b 的两部分 P 和 Q，作为每次计算子密钥的输入。

表 10.8　DES 密钥变换缩小换位 1

K(64b)→ P(28b)＋Q(28b)
P：56,48,40,32,24,16,8,0,57,49,41,33,25,17,9,1,58,50,42,34,26,18,10,2,59,51,43,35
Q：62,54,46,38,30,22,14,6,61,53,45,37,29,21,13,5,60,52,44,36,28,20,12,4,27,19,11,3

16 次密钥变换(i=0～15)的计算方式相同：对 P 和 Q 分别进行 $M[i]$位循环左移，M=\{1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1\}，得 P'和 Q'，合并得到 56b 后，再实施缩小选择换位 2(如表 10.9 所示)运算，即获得用于第 i 轮迭代运算的 48b 子密钥 K_i。

表 10.9　DES 密钥变换缩小换位 2

56b(P'＋Q')→ 48b(K_i)																							
13	16	10	23	0	4	2	27	14	5	20	9	22	18	11	3	25	7	15	6	26	19	12	1
40	51	30	36	46	54	29	39	50	44	32	47	43	48	38	55	33	52	45	41	49	35	28	31

DES 解密运算方法和加密完全一致，只是用相反的顺序使用子密钥。

在 DES 诞生之初，曾估计要耗资 2000 万美元的专用计算机连续运行 12 小时才可能破解 DES，因此当时被认为很强大。然而，这一经典的算法如今面临安全性严重不足的问题，一方面是破解方法不断被探索，另一方面主要是 56b 密钥长度太短，很难抵抗性能越来越强大的计算机进行的暴力攻击，可能只需价值数千美元的计算机就可在短时间内找到密钥。

改善 DES 保密性的一种可行方式为**三重 DES**(Triple-DES，3DES)，即对一个明文分组用 3 个相同或不同的密钥实施 3 次 DES 加密(或解密)运算，可获得大约相当于 112b 长度密钥的强度。3DES 方法如下。

设加密操作为 E，解密操作为 D，密钥为 K_1、K_2、K_3，则可采用以下四种模型之一实现加密：

$E(K_1)E(K_2)E(K_3)$；$E(K_1)D(K_2)E(K_3)$；$E(K_1)E(K_2)E(K_1)$；$E(K_1)D(K_2)E(K_1)$。

3DES 的解密方式是加密的逆过程。例如，对 $E(K_1)D(K_2)E(K_3)$模型的加密，解密应

为 $D(K_3)E(K_2)D(K_1)$。

DES 加密实际应用时,按照在加密过程中对明文分组不同的处理手段,DES 有如下 4 种工作方式(如图 10.11 所示)可供选择。设明文被分为 n 个分组,明文和密文分组分别用 m_i 和 c_i 表示。

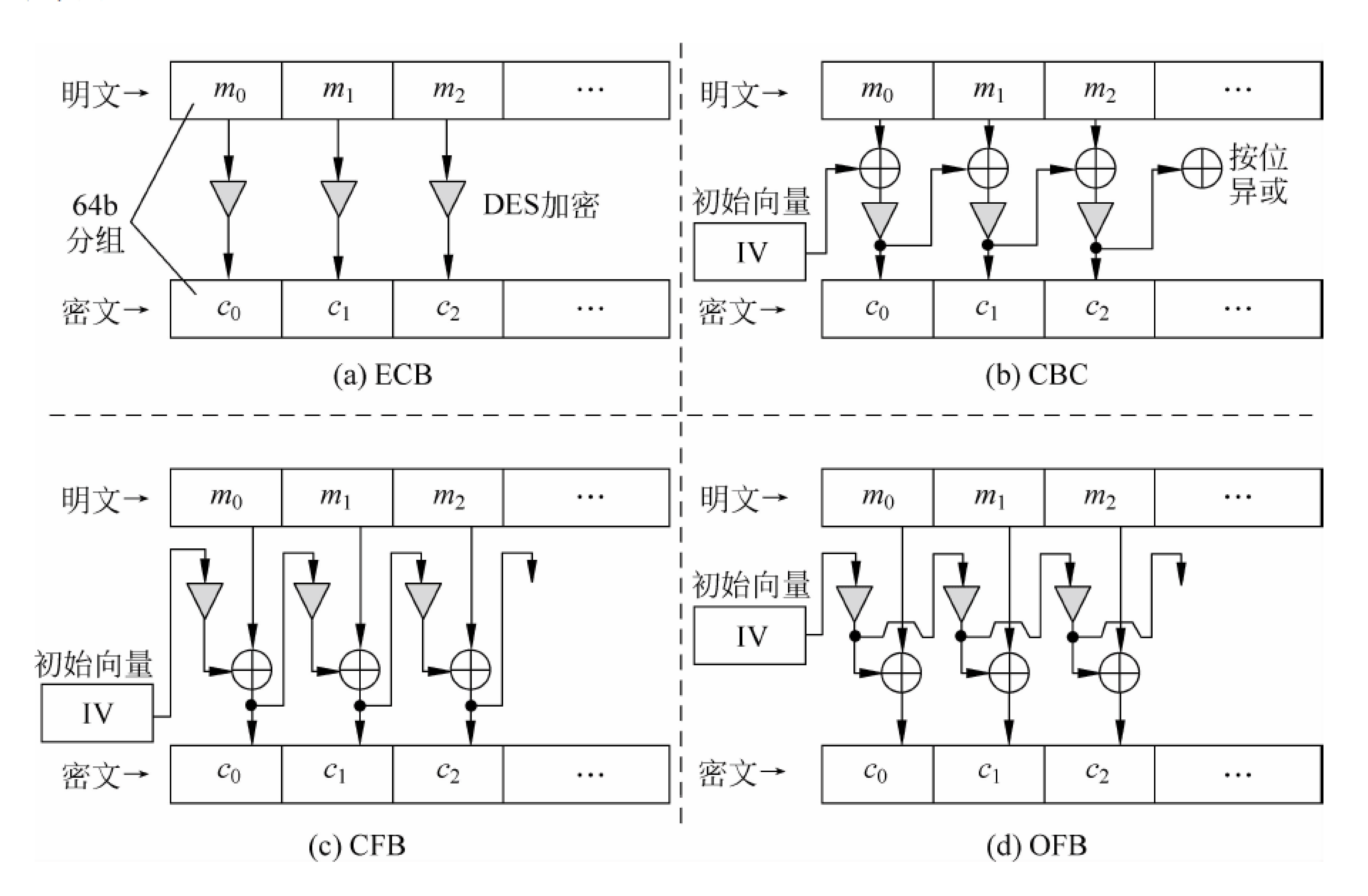


图 10.11 DES 的 4 种工作方式

(1) **电子密码本**(Electronic Code Book,ECB)。ECB 是最普通的 DES 加密方式,将明文分成 64b 分组,用密钥 K 分别进行加密(如图 10.11(a)所示)。ECB 加密和解密关系式分别为

$$c_i = E(m_i), \quad i = 0,1,\cdots,n-1; \; m_i = D(c_i), \quad i = 0,1,\cdots,n-1$$

ECB 方式适合于并行处理,但相同的明文分组会得到相同的密文,若被成组替换或修改难以发现。

(2) **密文分组链**(Cipher Block Chaining,CBC)。CBC 方式下每个明文分组与前一分组密文按位异或后再进行加密(如图 10.11(b)所示)。

第一个分组的加密需要一个初始值(Initialization Vector,IV),例如用一个随机数。选择的随机数应当随密文告知接收(解密)方。CBC 加密和解密关系式分别为

$$c_i = E(m_i \oplus c_{i-1}), \quad i = 0,1,\cdots,n-1, c_{-1} = \text{IV}$$

$$m_i = D(c_i) \oplus c_{i-1}, \quad i = 0,1,\cdots,n-1, c_{-1} = \text{IV}$$

CBC 方式克服了 ECB 方式的缺点,使相同的明文分组加密后得到不同的密文。但由于明文分组加密与前一组密文有关,因此加密和解密操作均无法并行处理,而且一个密文分组的数据传输错误会传播(影响)到下一分组的解密(思考为什么)。

(3) **密文反馈**(Cipher Feed Back,CFB)。CFB 方式使每个明文分组与前一分组密文的加密值进行按位异或来作为密文(如图 10.11(c)所示)。对第一个分组加密需设置初始值(IV)。CFB 加密和解密关系式分别为

$$c_i = E(c_{i-1}) \oplus m_i, \quad i = 0,1,\cdots,n-1, c_{-1} = \text{IV}$$

$$m_i = E(c_{i-1}) \oplus c_i, \quad i = 0,1,\cdots,n-1, c_{-1} = \mathrm{IV}$$

CFB 加密与前一分组密文相关，因此无法进行并行处理（但解密可并行处理），而且错误会传播到下一组。更大的问题是，由于密文仅仅是明文与一个数值的异或结果，当明文为易推测的文本型数据时，容易遭到破解（思考如何破解）。

（4）**输出反馈**（Output Feed Back，OFB）。OFB 方式与 CFB 相似，唯一不同的是 OFB 是直接取自前一分组 DES 的输出。OFB 加密和解密关系式分别为

$$c_i = E^{i+1}(\mathrm{IV}) \oplus m_i, \quad i = 0,1,\cdots,n-1; \; m_i = E^{i+1}(c_{i-1}) \oplus c_i, \quad i = 0,1,\cdots,n-1$$

虽然 OFB 的加密和解密操作都无法并行处理，但每个分组解密操作相互独立（与前一块密文无关），所以，只要 IV 传输不出错，就能克服 CFB 密文错误传播的缺点。

实际上，CFB 和 OFB 拥有更一般化的模型，具有可变长分组加密的能力，实用性更强，也更有意义。以 CFB 为例：设 64b 移位寄存器 R_j，可执行逻辑左移 jb 的操作，移位后的右侧 jb（最低 jb）可存入新的 jb 数据；再设 64b 选择寄存器 S_j，用于选择左侧 jb 的操作（取最高 jb 为有效数据，丢弃或忽略其余部分）。

如图 10.12 所示，OFB 可变长分组加密模型可指定加密 jb 明文分组，产生 jb 密文分组。加密过程为：将 64b 的 R_j 寄存器进行 DES 加密后，结果存入 S_j 寄存器；取 S_j 的最高 jb，与明文 jb 分组 m_i 异或，即得到 jb 密文 c_i；将前一 R_j 寄存器逻辑左移 jb 后，最低 jb 存入密文 c_i（OFB 方式则存入 S_j 的最高 jb），随后进入下一分组的加密过程。第 1 个分组的加密需要使用初始向量 IV。

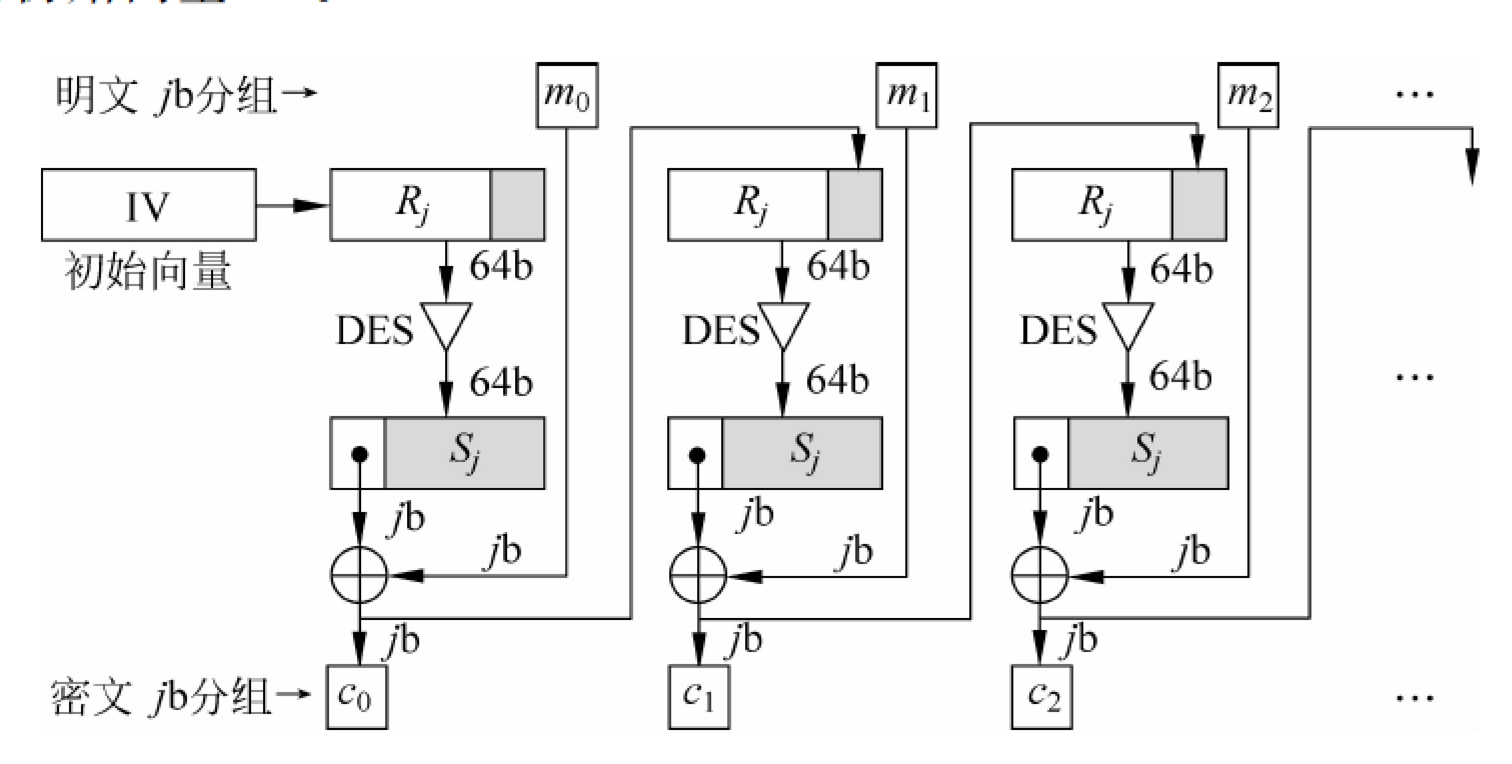


图 10.12　OFB 可变长加密模型

加密和解密关系式分别为（式中 R_i 表示寄存器 R 第 i 步时的值，$i=0,1,\cdots,n-1$）

$$\begin{cases} c_i = S_j[E(R_i)] \oplus m_i \\ R_i = (R_{i-1} \ll j) + c_{i-1} \\ R_0 = \mathrm{IV} \end{cases}$$

$$\begin{cases} m_i = S_j[E(R_i)] \oplus c_i \\ R_i = (R_{i-1} << j) + c_{i-1} \\ R_0 = \mathrm{IV} \end{cases}$$

考虑当 $j=8$ 时，OFB 即实现了以字节为分组单元进行 DES 加密的功能，从而成为一种流式加密的方法。

10.3.7 AES 算法

高级加密标准(Advanced Encryption Standard,AES)的基础是 2000 年 10 月 2 日由比利时学者 Joan Daemen 和 Vincent Rijmen 提出的 Rijndael 算法,2001 年 11 月 26 日被正式确定为新一代的美国联邦数据加密标准。

AES 可采用三种密钥长度:128b、192b 和 256b。理论上,AES 的 128b 密钥比 DES 的 56b 密钥强 1000 多倍。

AES 属于对称密钥分组加密算法,每个分组块为 128b。数据分组加密的轮次数与密钥长度有关,分别为 10、12 和 14 轮;每一轮都需要一个与输入分组具有相同长度的扩展密钥(子密钥);算法中采用一个密钥扩展程序,把输入密钥扩展成更长的比特串,以生成各轮的加密和解密密钥。

如图 10.13 所示,以 128b 密钥为例,AES 加密和解密运算进行 10 轮,每一轮有相似的迭代流程(其中 9 轮完全相同),运算类型有轮密钥加、字节代换、行移位、列混淆。解密运算逆向使用子密钥。

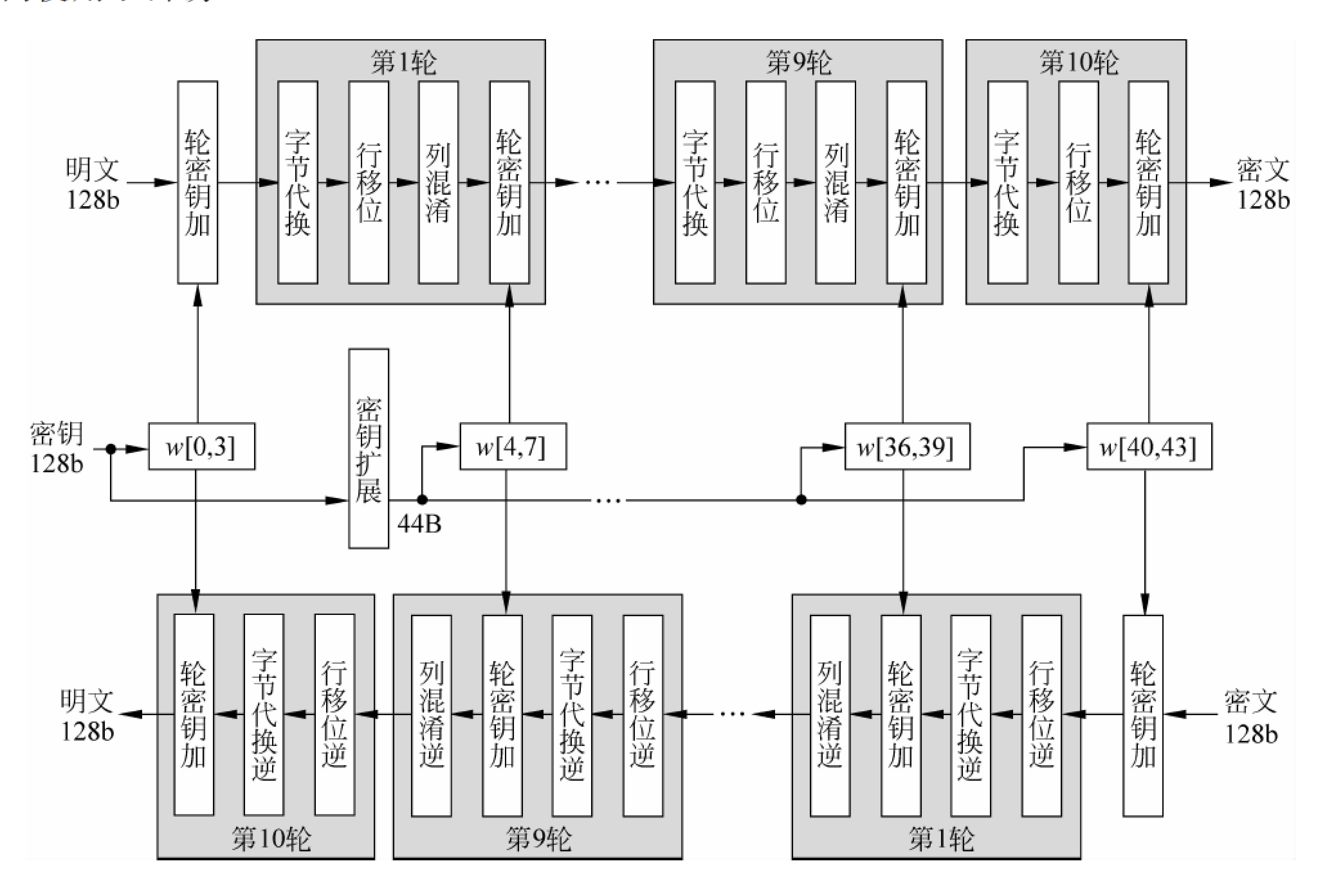


图 10.13 AES 加密和解密流程

AES 算法具有快速、高效、易用、灵活的特点,安全保密性强,应用领域很广,是 DES 算法的替代技术之一。

10.3.8 IDEA

国际数据加密算法(International Data Encryption Algorithm,IDEA)是一种对称密钥分组加密算法,基于相异代数群上的混合运算理论,由 Xujia Lai 和 James Massey 于 1990 年发布,后又发布了改进版算法以抵御差分攻击法,于 1992 年定名为 IDEA。

IDEA 使用长度为 128b 的密钥，数据分组大小为 64b。

IDEA 执行步骤如表 10.10 所示。其中：第 i 轮($i=0,1,\cdots,7$)迭代中需要的 6 个子密钥记为 $K_{6i+1}\sim K_{6i+6}$；$K_{49}\sim K_{52}$ 为最后 4 个子密钥；采用的 16b 数值运算有＋为模 2^{16} 加法、×为模($2^{16}+1$)乘法、⊕为按位异或。

表 10.10　IDEA 执行步骤

明文	$\{X_1, X_2, X_3, X_4\}$	64b
轮输入	X_1；X_2；X_3；X_4	共 8 轮，每轮输出为下一轮的输入
第 i 轮迭代 $i=0$，$1,\cdots,7$	(1) $Y_1=X_1\times K_{6i+1}$；(2) $Y_2=X_2+K_{6i+2}$；(3) $Y_3=X_3+K_{6i+3}$；(4) $Y_4=X_4\times K_{6i+4}$；(5) $Y_5=Y_1\oplus Y_3$；(6) $Y_6=Y_2\oplus Y_4$；(7) $Y_7=Y_5\times K_{6i+5}$；(8) $Y_8=Y_6+Y_7$；(9) $Y_9=Y_8\times K_{6i+6}$；(10) $Y_{10}=Y_7+Y_9$；(11) $Y_{11}=Y_1\oplus Y_9$；(12) $Y_{12}=Y_3\oplus Y_9$；(13) $Y_{13}=Y_2\oplus Y_{10}$；(14) $Y_{14}=Y_4\oplus Y_{10}$	
轮输出	$X_1=Y_{11}$；$X_2=Y_{13}$；$X_3=Y_{12}$；$X_4=Y_{14}$	
最终变换	$X_1=X_1\times K_{49}$；$X_2=X_2+K_{50}$；$X_3=X_3+K_{51}$；$X_4=X_4\times K_{52}$	4×16b
密文	$\{X_1, X_2, X_3, X_4\}$	64b

IDEA 加密过程为：输入的 64b 分组被分成 4 个 16b 子分组 X_1、X_2、X_3 和 X_4，成为算法第一轮的输入。算法总共进行 8 轮。在每一轮中，4 个子分组和 6 个 16b 子密钥进行 14 步运算；在每轮之间，输出的第 2 和第 3 个子分组交换位置；最后，在输出前的最终变换中，4 个子分组与 4 个子密钥再进行运算。

IDEA 算法总共需要 52 个子密钥(8 轮中的每一轮需要 6 个，最后 4 个用于输出变换)。子密钥的产生方法是：首先，将 128b 密钥 K 分成 8 个 16b 子密钥 $K_1\sim K_8$，作为算法的第一批 8 个子密钥(第一轮的 6 个，第二轮的头 2 个)；然后，整个密钥循环左移 25b 后，再分成 8 个子密钥 $K_9\sim K_{16}$，4 个用于第二轮的后 4 个，另外 4 个用在第三轮；依此类推，直到算法结束。

IDEA 解密过程与加密完全一样，子密钥的生成方法也相同，但子密钥的使用顺序和方式不同，需要进行数学变换，成为解密子密钥，并几乎逆序使用。

设 Z 为 16b 整数，则称 Z^{-1} 为 $Z \bmod (2^{16}+1)$的乘法逆元，全 0 用 2^{16} 代替，若以运算符⊠表示 mod ($2^{16}+1$)乘法，则有

$$Z \boxtimes Z^{-1} = 1\bmod(2^{16}-1) \quad 或 \quad Z^{-1} = \frac{1}{Z}\bmod(2^{16}-1)$$

又称 $-Z$ 为 $Z \bmod 2^{16}$ 的加法逆元(相当于取负)，若以运算符⊞表示 mod 2^{16} 加法，有 $Z \boxplus -Z=0 \bmod 2^{16}$。

记第 r 轮使用的第 i 个加密子密钥为 $k_i^{(r)}$，解密子密钥为 $d_i^{(r)}$($i=1,\cdots,6,r=1,\cdots,9$)，其中第九轮指最终变换。那么，IDEA 解密所使用的 52 个子密钥对应变换如下：

$$(d_1^{(r)},d_2^{(r)},d_3^{(r)},d_4^{(r)}) = (k_1^{(10-r)^{-1}}, -k_2^{(10-r)}, -k_3^{(10-r)}, k_4^{(10-r)^{-1}}),\quad r=1,9$$

$$(d_1^{(r)},d_2^{(r)},d_3^{(r)},d_4^{(r)}) = (k_1^{(10-r)^{-1}}, -k_3^{(10-r)}, -k_2^{(10-r)}, k_4^{(10-r)^{-1}}),\quad r=2,3,\cdots,8$$

$$(d_5^{(r)},d_6^{(r)}) = (k_5^{(9-r)},k_6^{(9-r)}), r=1,2,\cdots,8$$

在计算解密子密钥时，求加法逆元很简单，只需用 2^{16} 作被减数进行减法运算，或取反加 1。求乘法逆元的运算则具有一定的复杂性。可采用**扩展欧几里德算法**，有利于计算机执行。设求解 $k^{-1} \bmod n$，算法 ExtendedEuclid(k, n)流程如下：

```
1 (X1, X2, X3) := (1, 0, n)
2 (Y1, Y2, Y3) := (0, 1, k)
3 if (Y3 = 0) then return failure              //0 不存在乘法逆元
4 if (Y3 = 1) then return Y2                   //Y2 即为 k 的乘法逆元
5 Q := X3 div Y3                               //div 为除法取商
6 (T1, T2, T3) := (X1 - Q * Y1, X2 - Q * Y2, X3 - Q * Y3)
7 (X1, X2, X3) := (Y1, Y2, Y3)
8 (Y1, Y2, Y3) := (T1, T2, T3)
9 goto 3                                       //循环求解
```

CHAPTER 11

第11章 非对称密钥加密

11.1 非对称密钥加密原理

非对称密钥加密(Asymmetric Key Cryptography)也称**公开密钥加密**(Public Key Cryptography)或**公钥加密**、**双密钥加密**,是1976年由Diffie和Hellman在其《密码学新方向》一文中提出的技术。公钥加密方法使用一对密钥来加密和解密,其中一个是只有密钥拥有者自己知道的保密的**私钥**(private key),另一个是通信过程中由对方使用的**公钥**(public key)。

公钥体制的优越性在于具有两个相关的密钥,且其中的公钥是可以公开的,而私钥绝对不在网络上传输,因此就不存在密钥泄露问题。

用公钥加密的数据只能用对应的私钥解密,而用私钥加密的数据只能用对应的公钥解密。由于公钥加密技术的效率通常很低,甚至只有对称加密方法的千分之一,因此不适合对大量的数据进行加密,一般用来加密会话密钥等短小数据。

利用非对称密钥加密技术,可以实现三种不同安全效果的应用。

(1) 公钥加密-私钥解密。发送方使用数据通信的接收方提供的公钥来加密数据。因为只有合法的接收者才握有能正确解密的私钥,所以这种方法可以用来安全地传输保密数据。但是,假如攻击者使用了公钥(因为公钥是容易获得的)给接收方传送虚假的加密数据,接收者无法甄别该数据的来源是否合法(来自发送方),即发送者的身份无法被认定,发送方也因此是可抵赖的。

(2) 私钥加密-公钥解密。发送方使用自己拥有的私钥来加密数据并传送,接收方使用发送方提供的公钥来解密数据。由于只有发送方才能实施加密操作,因此接收方可以据此确定发送者的身份。然而,攻击者同样能够获得公钥,因而也能解密密文,所以这种方法不宜用来保护秘密数据。

(3) 公钥和私钥综合方法。发送方将对方的公钥和自己的私钥结合起来使用,就可达到数据保密通信、数据来源可确认(发送方不可否认)的双重目

的。设Alice需要发送机密信息给Bob，且Bob需要确认信息来自Alice，且Alice不能抵赖，则算法流程如下。

① Alice：生成密钥对 pub-K_a 和 pri-K_a，将公钥 pub-K_a 交给Bob。

② Bob：生成密钥对 pub-K_b 和 pri-K_b，将公钥 pub-K_b 交给Alice。

③ Alice：用Bob的公钥 pub-K_b 加密明文 M 得到密文 D_1，再用自己的私钥 pri-K_a 加密 D_1 得到密文 D_2，发送 D_2 给Bob。

④ Bob：先用Alice的公钥 pub-K_a 解密 D_2 得到 D_1，然后用自己的私钥 pri-K_b 解密 D_1 得到明文 M。

思考：该“综合方法”存在什么风险？发送方可否先用自己的私钥加密，再用对方的公钥加密？

考察古典加密方法Hill密码的加密和解密运算。设明文矩阵为 $\boldsymbol{P}$，密文矩阵为 $\boldsymbol{C}$，加密密钥矩阵为 $\boldsymbol{K}$，则根据Hill算法：$\boldsymbol{C}=\boldsymbol{K}\times\boldsymbol{P}$ 和 $\boldsymbol{P}=\boldsymbol{K}^{-1}\times\boldsymbol{C}$。

如果将Hill密钥矩阵 $\boldsymbol{K}$ 和 $\boldsymbol{K}^{-1}$ 分别看做加密密钥和解密密钥，两个密钥是相关的(矩阵互逆关系)，则可以将其中一个密钥作为公钥，另一个作为私钥。虽然矩阵求逆很容易被破解，这种所谓的公钥加密方法没有实用意义，但这个例子至少说明了公钥体制是完全可行的。

11.2 非对称密钥加密算法

11.2.1 RSA算法

RSA算法以其三位发明者Ron Rivest、Adi Shamir和Leonard Adleman名字的首字母来命名，是最早也是最著名的公开密钥加密算法，应用相当广泛。RSA算法的数学基础是数论中的欧拉(Euler)定理，并建立在大整数分解质数(素数)因子的困难性之上。

RSA密钥生成过程如下。

(1) 选择不同的质数 p 和 q，计算 $n=p\times q$ 和 $\varphi(n)=(p-1)\times(q-1)$。

(2) 选择整数 e，与 $\varphi(n)$ 互质，且 $1<e<\varphi(n)$。

(3) 计算 d，使 $d\times e=1 \bmod \varphi(n)$。

(4) 抛弃 p 和 q，得：公钥为 $K_{pub}=\{e,n\}$，私钥为 $K_{pri}=\{d,n\}$。

其中，e 与 $\varphi(n)$ 互质意味着两个数的最大公因数(Greatest Common Divisor，GCD)为1。可以运用辗转相除法，又名**欧几里德算法**(Euclidean)，在尝试 e 是否与 $\varphi(n)$ 互质时，不需要先把两个数作质因数分解，即可求出最大公因数。求GCD的递归算法如下(不失一般性，设 $p>q$)：

```
int gcd(p, q)
{
    if ( (p mod q) < > 0 )
        return gcd( q, (p mod q) );
    else
        return q;
}
```

计算 d 实际上是求 e 的 mod $\varphi(n)$ 乘法逆元，可使用 IDEA 中用到的扩展欧几里德算法。

使用公钥 $K_{pub}=\{e,n\}$ 的加密过程如下：对于明文 M（若 $M<n$，将 M 作为一个大整数；若 $M\geqslant n$，可分段加密）：

$$C = M^e \bmod n$$

使用私钥 $K_{pri}=\{d,n\}$ 的解密过程如下：

$$M = C^d \bmod n$$

解密与加密为互逆的运算。虽然非授权窃听者可以得到 e 和 n，但难以分解大数 n 以获取至关重要的 p 和 q，从而无法获得 d，因此无法解密。

例如，选择质数 $p=101$ 和 $q=113$，有

$$n = p \times q = 11\,413;\ \varphi(n) = (p-1)\times(q-1) = 100\times 112 = 11\,200$$

用欧几里德辗转相除法可求得 e，假设求得 $e=3533$，得到公钥 $K_{pub}=\{3533,\ 11\,413\}$。再用扩展欧几里德算法求 d，可得 $d=6597(\text{mod } 11\,200)$，得到私钥 $K_{pri}=\{6597,\ 11\,413\}$。

如果用公钥加密并发送明文 9726，加密运算为

$$9726^{3533} \bmod 11\,413 = 5761$$

可在网络上发送密文 5761。当接收方收到密文 5761 时，用拥有的私钥进行解密即成功获得明文。解密运算为

$$5761^{6597} \bmod 11\,413 = 9726$$

例子中的 p 和 q 选择很小，是为了便于示范的需要。为达到一定的安全保密等级，实际应用中应选择 100 位左右的十进制质数，n 的长度至少要达到 512b。为了抵抗整数分解算法，对 p 和 q 另有如下要求。

（1）$|p-q|$ 很大，通常 p 和 q 的长度相同。

（2）$p-1$ 和 $q-1$ 分别含有大素因子 p_1 和 q_1。

（3）p_1-1 和 q_1-1 分别含有大素因子 p_2 和 q_2。

（4）$p+1$ 和 $q+1$ 分别含有大素因子 p_3 和 q_3。

11.2.2　ElGamal 算法

ElGamal 算法是一种公开密钥加密技术，其安全性原理依赖于计算有限域上离散对数这一难题。ElGamal 密钥对的产生办法如下。

首先选择一个素数 p 和两个随机数 g 和 x（$g,x<p$），计算：

$$y = g^x \bmod p$$

则公钥为 $\{y,g,p\}$，私钥为 x。其中 g 和 p 可由一组用户共享。

ElGamal 进行公钥加密操作时，设明文为 M，需选择一个随机数 k，使 k 与 $(p-1)$ 互质（采用欧几里德辗转相除法），并计算

$$c_1 = g^k \bmod p$$

$$c_2 = y^k M \bmod p$$

所得 (c_1,c_2) 即为密文，是明文的两倍长度。解密时计算

$$M = \frac{b}{a^x} \bmod p$$

可采用扩展欧几里德算法,先求 a^x 的 mod p 乘法逆元,再与 b 相乘,以降低计算难度。

在运用 ElGamal 时,质数 p 必须足够大,且 $p-1$ 应至少包含一个大质数因子,并保证 g 对于 $p-1$ 的大素数因子不可约。ElGamal 的一个不足之处是其密文长度为明文的两倍,增加了存储空间和传输带宽的占用。

DSS(Digital Signature Standard)技术中采用的 DSA(Digital Signature Algorithm)算法就是经 ElGamal 算法演变而来。

11.2.3 ECC

椭圆曲线加密算法(Ellipse Curve Cryptography,ECC)是基于椭圆曲线理论的公钥加密技术。在数学上,对椭圆曲线的性质和功能的研究已逾150年,但是在加密技术上的应用是在1985年由 Neal Koblitz 和 Victor Miller 首次提出。与其他建立在大质数因子分解困难性基础上的加密方法不同,ECC 利用椭圆曲线方程式的数学性质产生密钥,正向计算比较容易,反过来却非常困难。

与 RSA 方法相比,ECC 可以使用164b密钥产生一个安全级,相当于 RSA 方法的1024b密钥提供的保密强度,而且计算量较小,处理速度更快,存储空间和传输带宽占用较少,具有较大的技术优势。

1. 射影平面

定义平行线相交于无穷远点 P_∞,这样,平面上所有直线都统一为有唯一的交点。P_∞ 具有以下性质。

(1) 一条直线只有一个无穷远点,一对平行线有公共的无穷远点。

(2) 任何两条不平行的直线有不同的无穷远点(否则会造成两个交点)。

(3) 平面上全体无穷远点构成一条无穷远直线。

平面上全体无穷远点与全体平常点构成射影平面。

对普通平面上点(x,y),令 $x=\dfrac{X}{Z}$,$y=\dfrac{Y}{Z}$,$Z\neq0$,则投影为射影平面上的点$(X:Y:Z)$。如点(1,3)可投影为$(Z:3Z:Z)$,即可为(1:3:1)、(2.3:6.9:2.3)等。

对普通平面上的直线 $ax+by+c=0$,同样变换,得到对应于射影平面上的直线为 $aX+bY+cZ=0$。

对平行线 $aX+bY+c_1Z=0$ 和 $aX+bY+c_2Z=0$,易解得 $Z=0$,说明无穷远点 P_∞ 的坐标为$(X:Y:0)$。

2. 椭圆曲线

一条椭圆曲线是在射影平面上满足**威尔斯特拉斯方程**(Weierstrass)的所有点的集合:

$$Y^2Z+a_1XYZ+a_3YZ^2=X^3+a_2X^2Z+a_4XZ^2+a_6Z^3 \tag{11.1}$$

椭圆曲线方程是一个齐次方程,曲线上的每个点都必须是非奇异的(光滑的),即偏导数 $F_X(X,Y,Z)$、$F_Y(X,Y,Z)$ 和 $F_Z(X,Y,Z)$ 不能同时为0。

椭圆曲线的形状并非椭圆,例如,方程 $Y^2Z=X^3+XZ^2+Z^3$ 的曲线如图11.1(a)所示(转换为 $y^2=x^3+x+1$),方程 $Y^2Z=X^3-XZ^2$ 的曲线如图11.1(b)所示(转换为 $y^2=x^3-x$),显然两个方程都满足式(11.1)的形式。

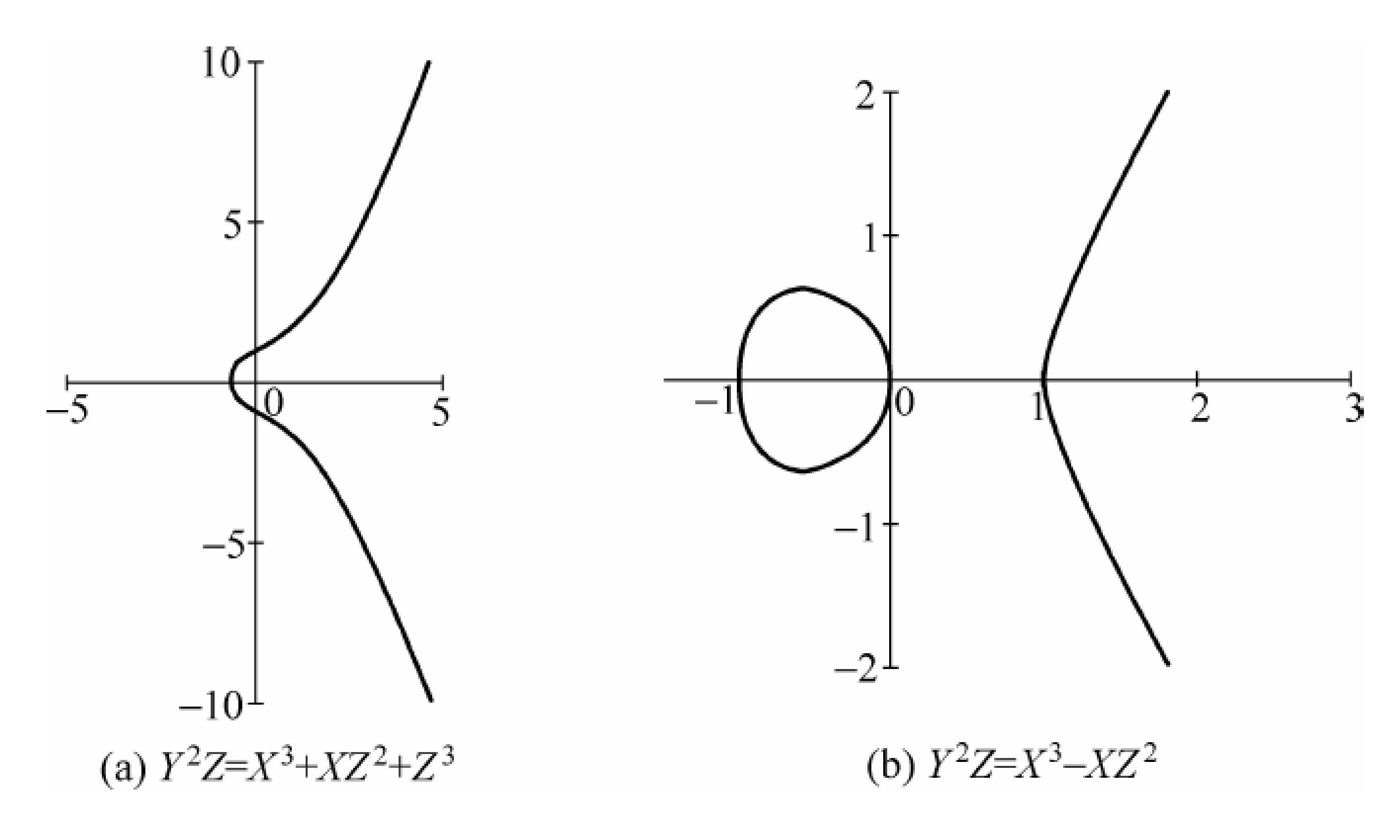


图 11.1　椭圆曲线示例

但并非所有形如式(11.1)的方程都是椭圆曲线方程,例如,图 11.2 所示的方程为 $Y^2Z=X^3+X^2$ 和 $Y^2Z=X^3$。

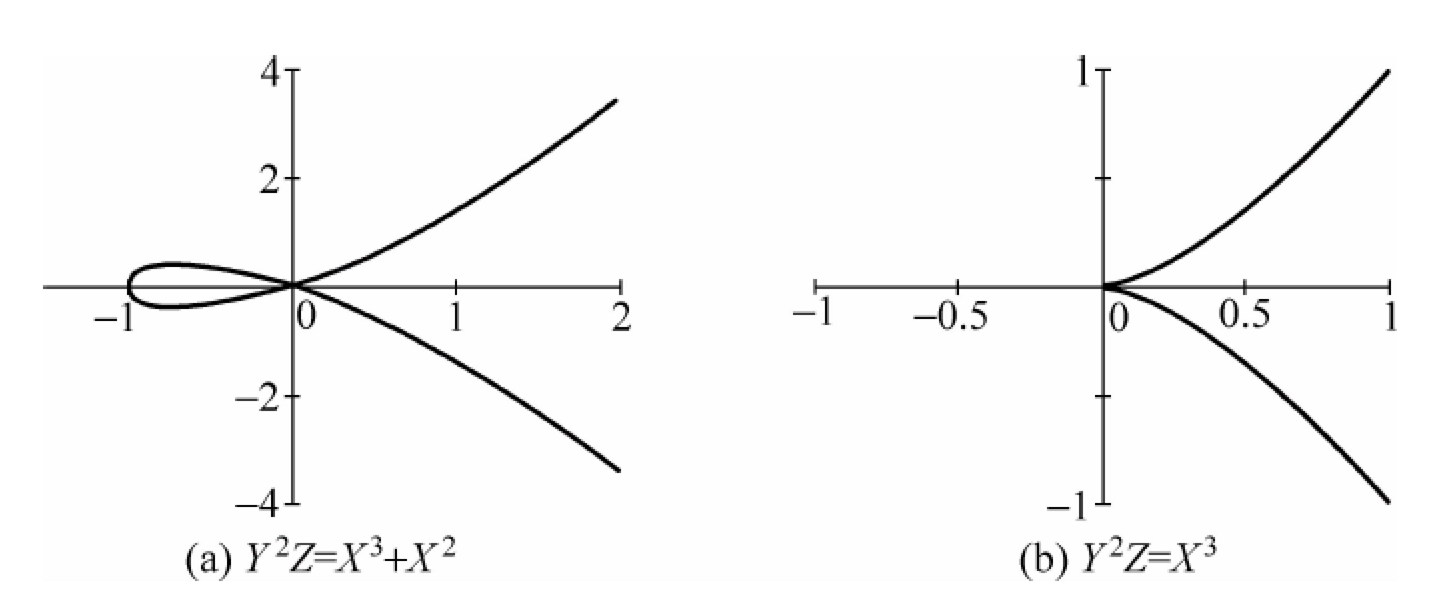


图 11.2　非椭圆曲线示例

当 $Z=0$,由式(11.1)知 $X=0$,则椭圆曲线上有一个无穷远点 O_∞,为$(0:Y:0)$。无穷远点和普通平面上的平常点(曲线)一起,即组成射影平面上的椭圆曲线。椭圆曲线方程对应的普通平面方程为

$$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6 \tag{11.2}$$

对式(11.2)曲线的平常点(x,y)求导,计算切线的斜率 k,有

$$F_x(x,y)=a_1y-3x^2-2a_2x-a_4$$

$$F_y(x,y)=2y+a_1x+a_3$$

$$k=f'(x)=-\frac{F_x(x,y)}{F_y(x,y)}=\frac{3x^2+2a_2x+a_4-a_1y}{2y+a_1x+a_3} \tag{11.3}$$

3. 椭圆曲线加法

将阿贝尔(Abel)加法群(又称交换群)的概念引入椭圆曲线。

任意取椭圆曲线上两点 P、Q(若 P、Q 两点重合,则作 P 点的切线),作直线交于椭圆曲线的另一点 R',过 R'作 y 轴的平行线交于 R,定义 $P+Q=R$。这样,加法的和也在椭圆曲线上,并同样具备加法的交换律、结合律。

例如,如图 11.1(b)所示的椭圆曲线方程为 $Y^2Z=X^3-XZ^2$,普通方程为 $y^2=x^3-x$,图 11.3 分别示意了 P、Q 重合与不重合两种情况。

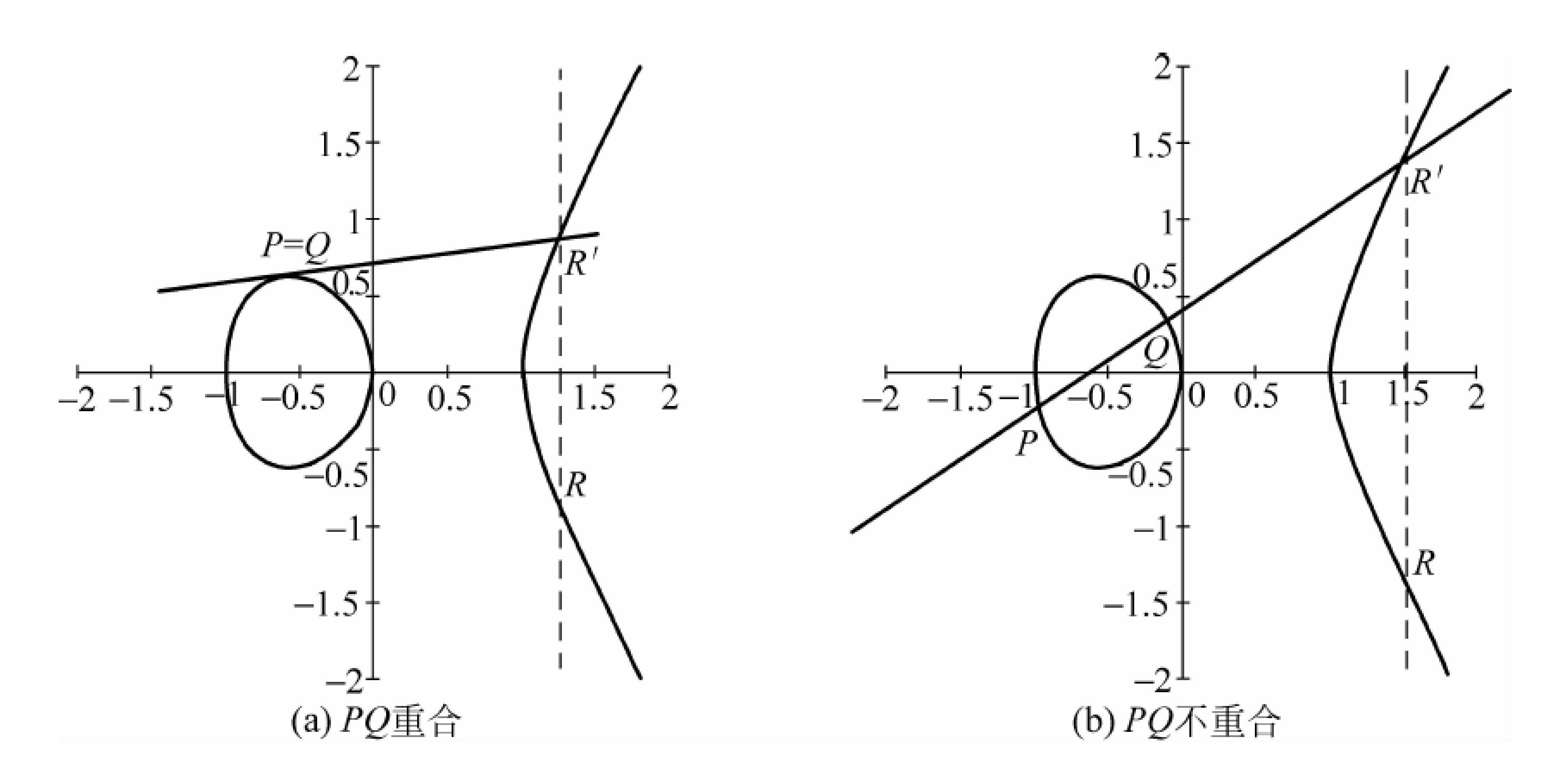


图 11.3 椭圆曲线加法示意

如图 11.4(a)所示,椭圆曲线无穷远点 O_∞ 与椭圆曲线上一点 P 的连线交于另一点 P',过 P' 作 y 轴的平行线交于 P,根据加法定义,有 $O_\infty+P=P$。可见,无穷远点 O_∞ 与普通加法中零相当,因此把 O_∞ 称为零元。同时易知 $P+P'=O_\infty$,则 P' 称为 P 的负元,记作 $-P$。如图 11.4(b)所示,还可推出:如果椭圆曲线上的 3 个点 A、B、C 处于同一直线上,那么其和等于零元,即 $A+B+C=O_\infty$。

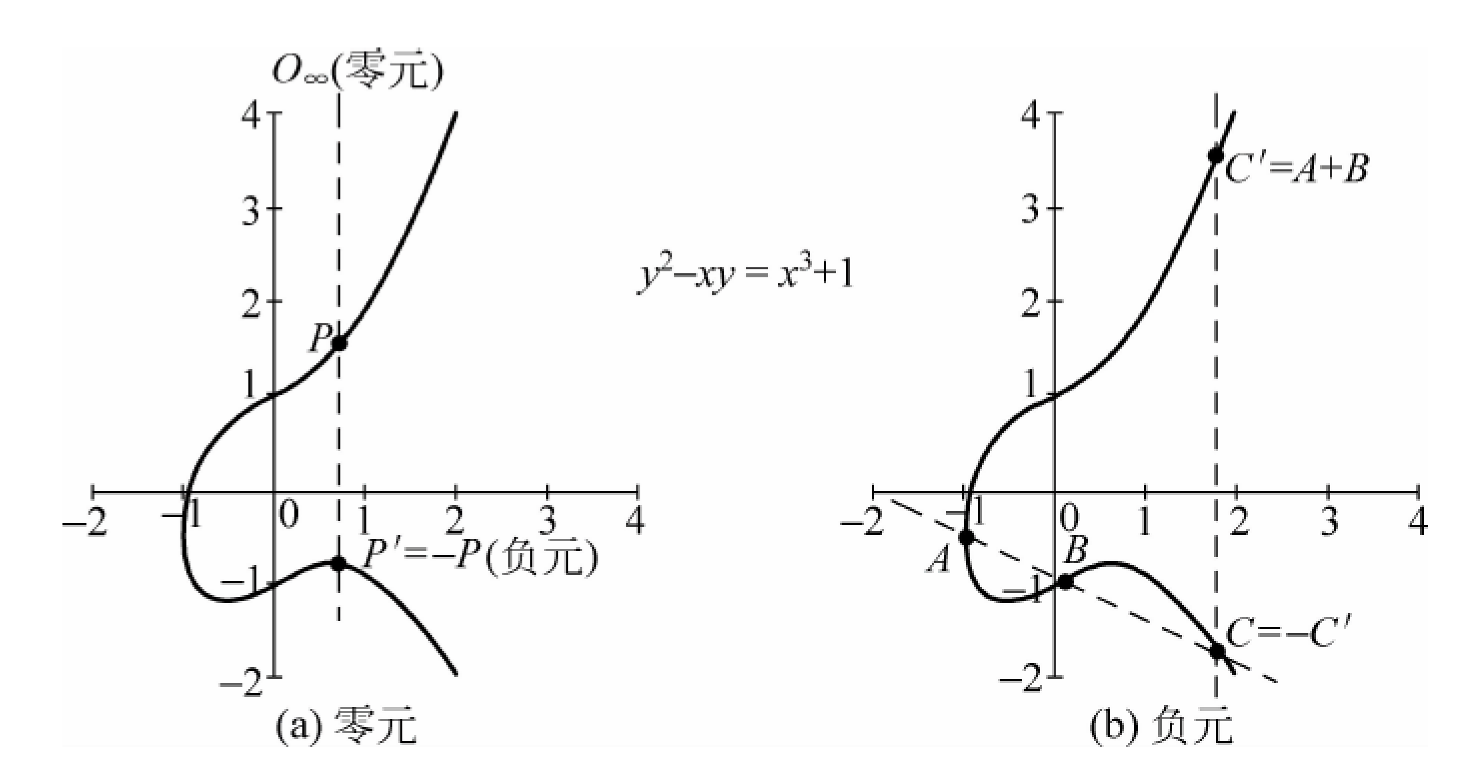


图 11.4 椭圆曲线零元与负元

若有 k 个相同的点 P 相加,记作 kP。如图 11.5 所示,有

$$P+P+P=2P+P=3P$$

在式(11.2)曲线上,若已知点 P、Q 的坐标分别为(x_1,y_1)、(x_2,y_2),令 $R=P+Q$,设:$-R$ 的坐标为(x_3,y_3),R 的坐标为(x_4,y_4)。

因为 P、Q、$-R$ 三点共线,所以设共线方程为 $y=kx+b$,其中:

(1) 若 $P\neq Q$(P、Q 两点不重合),则直线斜率为

$$k=\frac{y_1-y_2}{x_1-x_2}$$

(2) 若 $P=Q$(P、Q 两点重合),则直线为椭圆曲线的切线,因此由式(11.3)可知

$$k=\frac{3x_1^2+2a_2x_1+a_4-a_1y_1}{2y_1+a_1x_1+a_3}$$

因此，P、Q、$-R$ 三点的坐标值就是式(11.2)和共线方程组成的方程组的解。将共线方程代入式(11.2)后得

$$(kx+b)^2+a_1x(kx+b)+a_3(kx+b)$$
$$=x^3+a_2x^2+a_4x+a_6$$

将其化为一般方程，根据三次方程根与系数关系：当三次项系数为 1 时，$-x_1x_2x_3$ 等于常数项，$x_1x_2+x_2x_3+x_3x_1$ 等于一次项系数，$-(x_1+x_2+x_3)$ 等于二次项系数，有

$$-(x_1+x_2+x_3)=a_2-ka_1-k^2$$

则

$$x_4=x_3=k^2+ka_1-a_2-x_1-x_2$$

又由于 $k=\dfrac{y_1-y_3}{x_1-x_3}$，则

$$y_3=y_1-k(x_1-x_3)$$

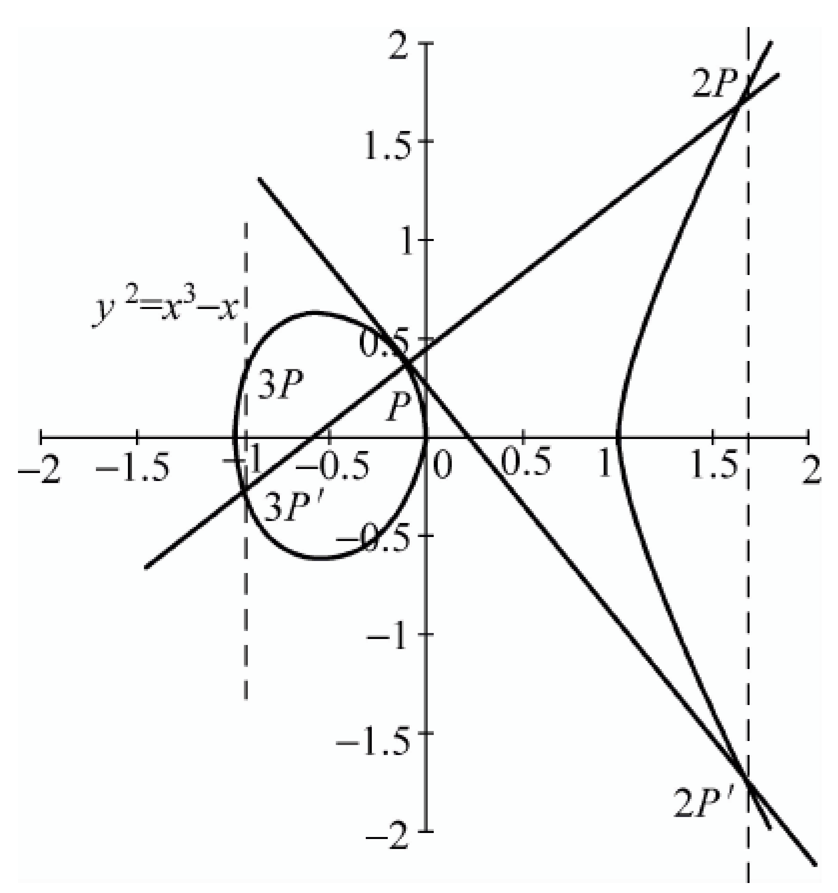


图 11.5　椭圆曲线同点加法示意

将 $x=x_4$ 代入式(11.2)，并化为一般方程，得

$$y^2+(a_1x_4+a_3)y-(x_4^3+a_2x_4^2+a_4x_4+a_6)=0$$

根据二次方程根与系数的关系，有 $-(a_1x_4+a_3)=y_3+y_4$，则

$$y_4=-y_3-(a_1x_4+a_3)$$

4. 有限域椭圆曲线

由于信息加密是在有限域上进行的，最大值、最小值由信息长度决定(并非无穷大)，而且是离散的整数，所以必须对实数域上的椭圆曲线进行改进。另外，椭圆曲线的选择很重要，因为并非所有椭圆曲线都适合加密。

给出有限域 Fp：

(1) Fp 中有 p(p 为质数)个元素 $0,1,2,\cdots,p-2,p-1$。

(2) Fp 的加法是 $a+b\equiv c(\bmod\ p)$。

(3) Fp 的乘法是 $a\times b\equiv c(\bmod\ p)$。

(4) Fp 的除法是 $a\div b\equiv c(\bmod\ p)$。

(5) Fp 的单位元是 1，零元是 0。

(6) Fp 域内运算满足交换律、结合律、分配律。

以椭圆曲线 $y^2=x^3+ax+b$ 为例，将其定义在 Fp 上，即满足下式的所有点(x,y)再加上无穷远点 O_∞。无穷远点 O_∞ 是零元，$O_\infty+O_\infty=O_\infty$，$O_\infty+P=P$。$P(x,y)$的负元是 $-P=P'(x,-y)$，有 $P+(-P)=O_\infty$。

$$y^2=x^3+ax+b\pmod p$$

选择质数 p，应有 $x,y\in[0,p-1]$。将这条椭圆曲线记为 $Ep(a,b)$。选择两个满足下列约束条件的小于 p 的非负整数 a、b：

$$4a^3+27b^2\neq 0\pmod p$$

当 $p=23$，$a=b=1$ 时，椭圆曲线 $E_{23}(1,1)$如图 11.6 所示。

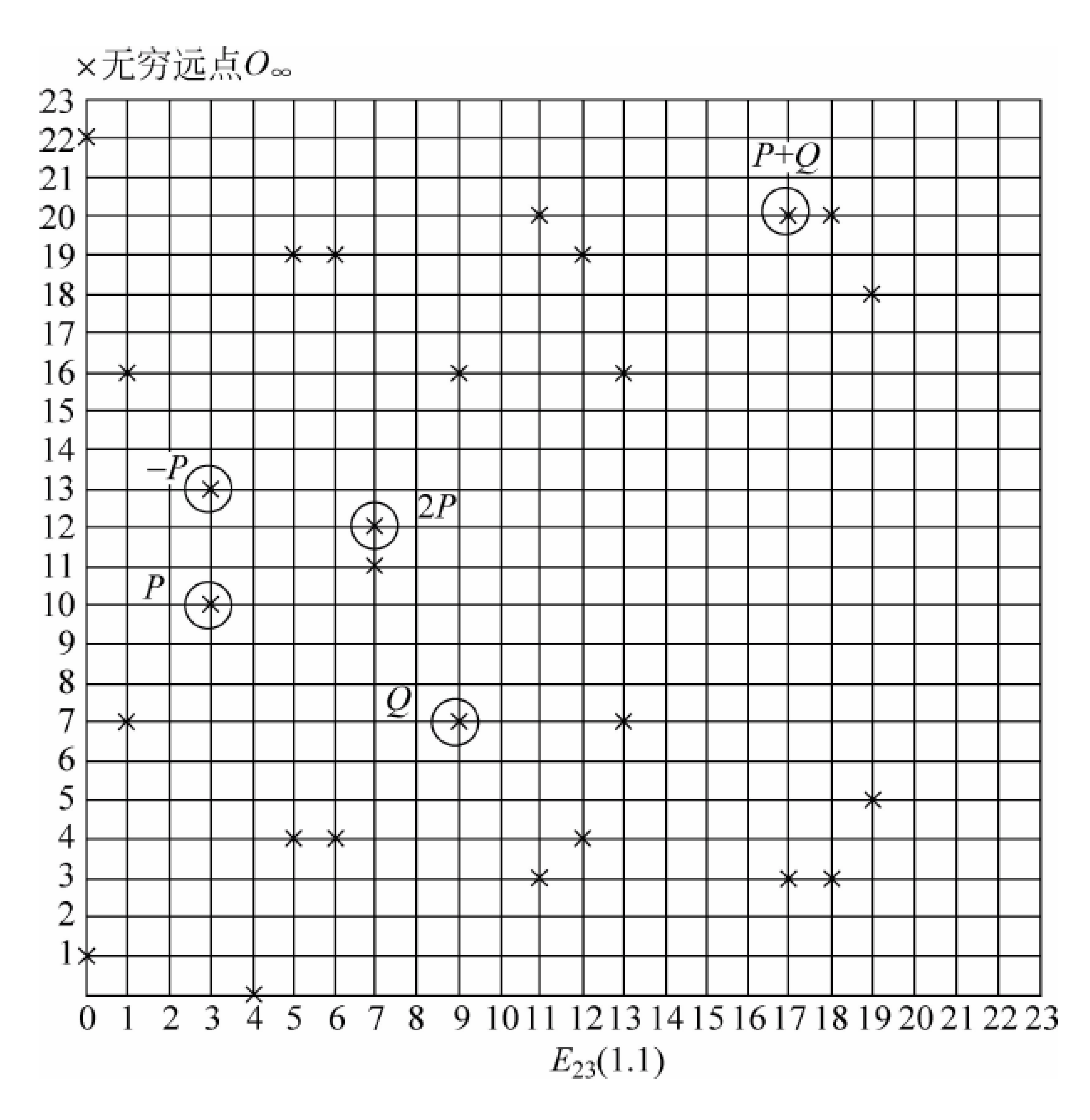


图 11.6　有限域椭圆曲线示例

如果已知两点 $P(3,10)$、$Q(9,7)$，则 $-P=(3,-10)$，即 $(3,13)$，$k=\dfrac{7-10}{9-3}=-\dfrac{1}{2}$，因为 $2\times12=1(\bmod 23)$，所以 2 的乘法逆元为 12，所以 $k\equiv-1\times12(\bmod 23)=11$，得

$$x=11^2-3-9(\bmod 23)=109(\bmod 23)=17$$
$$y=11\times(3-(-6))-10(\bmod 23)=89(\bmod 23)=20$$

因此 $P+Q$ 的坐标为 $(17,20)$。另外，$k'=\dfrac{3\times3^2+1}{2\times10}(\bmod 23)=\dfrac{1}{4}(\bmod 23)=6$，得

$$x=6^2-3-3(\bmod 23)=30(\bmod 23)=7$$
$$y=6\times(3-7)-10(\bmod 23)=-34(\bmod 23)=12$$

因此 $2P$ 的坐标为 $(7,12)$。

如果椭圆曲线上一点 P，存在最小的正整数 n，使得数乘 $nP=O_\infty$(显然 $(n-1)P=-P$)，则将 n 称为 P 的阶，若 n 不存在，则 P 是无限阶的。事实上，在有限域上定义的椭圆曲线上所有的点的阶 n 都是存在的。

5. ECC 加密方法

考虑 $K=kG$，其中 K、G 为椭圆曲线 $Ep(a,b)$ 上的点，n 为 G 的阶 $(nG=O_\infty)$，k 为小于 n 的整数。

不难发现，给定 k 和 G，根据加法法则，计算 K 很容易；但反过来，给定 K 和 G，求 k 就非常困难。这就是椭圆曲线加密算法的数学依据。

点 G 称为**基点**(base point)，$k(k<n)$ 为私钥，K 为公钥。

一个利用椭圆曲线算法进行保密通信(公钥加密-私钥解密)的典型过程可描述为：

(1) Alice 选定一条椭圆曲线 $Ep(a,b)$，并取椭圆曲线上一点作为基点 G。

(2) Alice 选择一个私有密钥 k，并生成公开密钥 $K=kG$。

(3) Alice 将 $Ep(a,b)$和点 K、G 传给 Bob。

(4) Bob 收到信息后，将待传输的明文编码到 $Ep(a,b)$上的一点 M(编码方法略)，并产生一个随机整数 $r(r<n)$。

(5) Bob 计算点 $C_1=M+rK$ 和 $C_2=rG$。

(6) Bob 将 C_1、C_2 传给 Alice。

(7) Alice 收到信息后，计算 C_1-kC_2，结果就应该是点 M，因为

$$C_1-kC_2=M+rK-k(rG)=M+r(K-kG)=M$$

在这个加密通信过程中，假定有一个窃密者 Henry，能够获取到 $Ep(a,b)$、K、G、C_1、C_2，然而通过 K、G 求 k，或通过 C_2、G 求 r 都是困难的，因此，Henry 无法解密 Alice 和 Bob 间传送的信息。

通常将 Fp 上的一条椭圆曲线描述为 $T=(p,a,b,G,n,h)$。其中：p、a、b 用来确定一条椭圆曲线，G 为基点，n 为点 G 的阶，h 是椭圆曲线上所有点的个数 m 与 n 相除的商的整数部分。

这 6 个参量取值的选择，直接影响了加密的安全性。参量值一般要求满足以下几个条件：p 越大安全性越好，但会导致计算速度变慢，200b 左右可满足一般安全要求；n 应为质数；$h\leqslant 4$；$p\neq n\times h$；$pt\neq 1(\mathrm{mod}\ n)$，$1\leqslant t<20$；$4a^3+27b^2\neq 0(\mathrm{mod}\ p)$。

CHAPTER 12

第 12 章　单向函数加密

12.1　单向函数加密原理

单向函数(One-way Function)又称为 **Hash 函数**、**哈希函数**、**散列函数**，可实施特殊的数据加密。

Hash 函数可把任意长度的输入变换成固定长度的输出，即原输入的 Hash 值，相当于一种压缩映射，因为 Hash 值的空间通常远小于输入值的空间。

考察**背包问题**(knapsack problem)：有一个背包以及 n 个物品，物品直径与背包相同，高度分别为 $a_1a_2\cdots a_n$。若已知各用 m_i 个正好装满背包，则有 $\sum a_im_i=b$，b 就是背包的高度；但是，若已知高度 b，问如何各选出 x_i 个物品正好装满背包，即求 $\sum a_ix_i=b$ 的解，显然是极其困难的。

设想把背包问题中的 $m_0m_1\cdots m_n$(设 $m_i\in\{0,1\}$)作为明文，$a_1a_2\cdots a_n$ 作为密钥，很容易计算出密文 b；但是，即使已知密钥，也难以从 b 反推出明文。Hash 算法即采用这一原理。

由于 Hash 值能够在统计上唯一地表征输入值，因此被称为**消息摘要**(Message Digest)，即把消息进行数学意义上的内容摘要，但从摘要中无法得到比摘要本身更多的关于原有消息的信息。Hash 算法需要满足以下关键特性。

(1) 单向性(one-way)。从输入值能够简单、迅速地得到 Hash 值，而反过来在计算上不可行。

(2) 抗冲突性(Collision Resistant)。给定 M，计算上无法找到 M'，满足 $H(M)=H(M')$，即弱抗冲突性；计算上也难以寻找一对任意的 M 和 M'，使满足 $H(M)=H(M')$，即强抗冲突性。

(3) 分布均匀性。存在映射分布均匀性和差分分布均匀性。Hash 值中，0 和 1 的数量应该大致相等；输入中每一个比特的变化，Hash 值中应有一半以上的比特改变，这被称为雪崩效应(avalanche effect)；要实现使 Hash

值中出现 1 比特的变化，则输入中至少有一半以上的比特必须发生变化。分布均匀性本质上是必须使输入中每一个比特的信息，尽量均匀地反映到输出的每一个比特上去；输出中的每一个比特，都是输入中尽可能多比特的信息一起作用的结果。

12.2　单向函数算法

12.2.1　MD5 算法

消息摘要(Message Digest，MD)是一种单向函数加密算法，分为 MD2、MD4 和 MD5，其中 MD5 算法最为常用。

MD5 以 512b(64B)分组为单位来处理输入信息。首先对信息进行填充，在信息的尾部填充一个 1 和连续的 0，直到满足比特长度对 512 求余的结果等于 448(即 $n\times512+448$)，其后附加 64b 原信息长度值(以 B 为单位)。

MD5 对 4 个 32b 的链接变量(chaining variable)分别设置初始值为：V_a=0x01234567，V_b=0x89abcdef，V_c=0xfedcba98，V_d=0x76543210。执行算法流程如下。

(1) 令 $a=V_a,b=V_b,c=V_c,d=V_d$。

(2) 将输入的 512b 分组划分成 16 个 32b 子分组 $m_j,j=0,1,\cdots,15$。

(3) 进行 4 轮运算，每轮由 16 步组成，每步为 $K(A,B,C,D,m_j,s,t_i)$函数运算(如图 12.1 所示)，K 函数表示为

$$A=B+((A+k(B,C,D)+m_j+t_i)\gg s)$$

① $K\in\{F,G,P,Q\}$，$k\in\{f,g,p,q\}$，$\gg s$ 表示循环右移 s 位，s 是常数(见图 12.1 中对应数值)。

② 分别用于四轮运算的 4 个非线性函数如下(每轮一个，均为按位逻辑运算：& 与，| 或，～非，⊕异或)：

$$f(x,y,z)=(x\&y)\mid((\sim x)\&z)$$
$$g(x,y,z)=(x\&z)\mid(y\&(\sim z))$$
$$p(x,y,z)=x\oplus y\oplus z$$
$$q(x,y,z)=y\oplus(x\mid(\sim z))$$

③ 在第 i 步中($i=1,2,\cdots,64$)，以 i 为弧度值，可计算出常数

$$t_i=\text{int}(2^{32}\times|\sin(i)|)$$

(4) 计算：$V_a=V_a+a,V_b=V_b+b,V_c=V_c+c,V_d=V_d+d$。

(5) 对下一个 512b 分组，回到(1)继续运算。

(6) 对所有原消息分组运算完毕后，最后输出 V_a、V_b、V_c、V_d 的级联$\{V_a,V_b,V_c,V_d\}$，即为 MD5 值。

例如，第二轮的第 1 步(总第 17 步)为 $G(a,b,c,d,m_1,s,t_{17})$，表示

$$a=b+((a+g(b,c,d)+m_1+t_{17})\gg s)$$

即为

$$a=b+((a+((b\&d)\mid(c\&(\sim d)))+m_1+t_{17})\gg s)$$

也即

$$a = b + ((a + ((b\&d) \mid (c\&(\sim d))) + m_1 + 0\text{xf61e2562}) \gg 5)$$

第一轮	第二轮
F(a,b,c,d,m0 , 7,0xd76aa478)	G(a,b,c,d,m1 , 5,0xf61e2562)
F(d,a,b,c,m1 ,12,0xe8c7b756)	G(d,a,b,c,m6 , 9,0xc040b340)
F(c,d,a,b,m2 ,17,0x242070db)	G(c,d,a,b,m11,14,0x265e5a51)
F(b,c,d,a,m3 ,22,0xc1bdceee)	G(b,c,d,a,m0 ,20,0xe9b6c7aa)
F(a,b,c,d,m4 , 7,0xf57c0faf)	G(a,b,c,d,m5 , 5,0xd62f105d)
F(d,a,b,c,m5 ,12,0x4787c62a)	G(d,a,b,c,m10, 9,0x02441453)
F(c,d,a,b,m6 ,17,0xa8304613)	G(c,d,a,b,m15,14,0xd8a1e681)
F(b,c,d,a,m7 ,22,0xfd469501)	G(b,c,d,a,m4 ,20,0xe7d3fbc8)
F(a,b,c,d,m8 , 7,0x698098d8)	G(a,b,c,d,m9 , 5,0x21e1cde6)
F(d,a,b,c,m9 ,12,0x8b44f7af)	G(d,a,b,c,m14, 9,0xc33707d6)
F(c,d,a,b,m10,17,0xffff5bb1)	G(c,d,a,b,m3 ,14,0xf4d50d87)
F(b,c,d,a,m11,22,0x895cd7be)	G(b,c,d,a,m8 ,20,0x455a14ed)
F(a,b,c,d,m12, 7,0x6b901122)	G(a,b,c,d,m13, 5,0xa9e3e905)
F(d,a,b,c,m13,12,0xfd987193)	G(d,a,b,c,m2 , 9,0xfcefa3f8)
F(c,d,a,b,m14,17,0xa679438e)	G(c,d,a,b,m7 ,14,0x676f02d9)
F(b,c,d,a,m15,22,0x49b40821)	G(b,c,d,a,m12,20,0x8d2a4c8a)
第三轮	第四轮
P(a,b,c,d,m5 , 4,0xfffa3942)	Q(a,b,c,d,m0 , 6,0xf4292244)
P(d,a,b,c,m8 ,11,0x8771f681)	Q(d,a,b,c,m7 ,10,0x432aff97)
P(c,d,a,b,m11,16,0x6d9d6122)	Q(c,d,a,b,m14,15,0xab9423a7)
P(b,c,d,a,m14,23,0xfde5380c)	Q(b,c,d,a,m5 ,21,0xfc93a039)
P(a,b,c,d,m1 , 4,0xa4beea44)	Q(a,b,c,d,m12, 6,0x655b59c3)
P(d,a,b,c,m4 ,11,0x4bdecfa9)	Q(d,a,b,c,m3 ,10,0x8f0ccc92)
P(c,d,a,b,m7 ,16,0xf6bb4b60)	Q(c,d,a,b,m10,15,0xffeff47d)
P(b,c,d,a,m10,23,0xbebfbc70)	Q(b,c,d,a,m1 ,21,0x85845dd1)
P(a,b,c,d,m13, 4,0x289b7ec6)	Q(a,b,c,d,m8 , 6,0x6fa87e4f)
P(d,a,b,c,m0 ,11,0xeaa127fa)	Q(d,a,b,c,m15,10,0xfe2ce6e0)
P(c,d,a,b,m3 ,16,0xd4ef3085)	Q(c,d,a,b,m6 ,15,0xa3014314)
P(b,c,d,a,m6 ,23,0x04881d05)	Q(b,c,d,a,m13,21,0x4e0811a1)
P(a,b,c,d,m9 , 4,0xd9d4d039)	Q(a,b,c,d,m4 , 6,0xf7537e82)
P(d,a,b,c,m12,11,0xe6db99e5)	Q(d,a,b,c,m11,10,0xbd3af235)
P(c,d,a,b,m15,16,0x1fa27cf8)	Q(c,d,a,b,m2 ,15,0x2ad7d2bb)
P(b,c,d,a,m2 ,23,0xc4ac5665)	Q(b,c,d,a,m9 ,21,0xeb86d391)

图 12.1　MD5 四轮函数运算表

MD5 加密字符串的实例如下：

```
md5 ("") = d41d8cd98f00b204e9800998ecf8427e
md5 ("a") = 0cc175b9c0f1b6a831c399e269772661
md5 ("abc") = 900150983cd24fb0d6963f7d28e17f72
md5 ("message digest") = f96b697d7cb7938d525a2f31aaf161d0
md5 ("abc…z") = c3fcd3d76192e4007dfb496cca67e13b
```

MD5 加密方法常用于信息系统用户口令的存储,防止非法窃取。但这种安全机制并非万无一失,攻击者不必费心解密 Hash 值,可采用字典法(明文串和 Hash 值对应表)、重放法

等实施破解。

12.2.2　SHA

安全散列算法(Secure Hash Algorithm,SHA)是单向函数加密算法之一,包括 SHA-1、SHA-224、SHA-256、SHA-384 和 SHA-512,1995 年发布为美国联邦标准。后四者并称为 SHA-2。SHA-1 在 SSL、SSH、S/MIME 和 IPSec 等许多安全协议中广为使用,被视为 MD5 的继任者。但与 MD5 的命运相似,SHA-1 的安全性如今已被质疑。虽然相比之下 SHA-2 更为安全,但其算法与 SHA-1 基本一致。

SHA-1 以 512b 分组(32b 字长)为处理单位,输出 160b 值;SHA-xyz 表示输出 xyzb 值,前两者分组大小与 SHA-1 相同为 512b(32b 字长),后两者为 1024b(64b 字长)。

SHA-1 算法如下(所有变量为 32b 字长,计算均为 mod 2^{32})。

对原消息的预处理与 MD5 算法相同:在原消息的尾部填充一个 1 和连续的 0,直到满足比特长度对 512 求余的结果等于 448(即 $n\times512+448$),其后附加 64b 原消息长度值(以 B 为单位)。

设置中间变量初始值为:h_0 :=0x67452301;h_1 :=0xefcdab89;h_2 :=0x98badcfe;h_3 :=0x10325476;h_4 :=0xc3d2e1f0。将原消息分为 512b 长度的分组(chunk);依次处理每个分组,直到处理完全部分组。

(1) 将分组分为 16 个 32b 字 $w[i]$,$i=0,1,\cdots,15$。

(2) 将 $w[i]$,$i=0,1,\cdots,15$ 扩展成 80 个 32b 字($<<$循环左移,$\oplus$异或):

```
for i from 16 to 79
    w[i] := (w[i-3]⊕w[i-8]⊕w[i-14]⊕w[i-16])<<1
```

(3) 变量赋值:$\{a,b,c,d,e\}=\{h_0,h_1,h_2,h_3,h_4\}$。

(4) 主处理程序(循环 80 轮次;& 与,| 或,~非):

```
for i from 0 to 79
    if 0 ≦ i ≦ 19 then
        f := (b& c) | ((~b) & d); k := 0x5a827999
    else if 20 ≦ i ≦ 39
        f := b⊕c⊕d; k := 0x6ed9eba1
    else if 40 ≦ i ≦ 59
        f := (b& c) | (b& d) | (c& d); k := 0x8f1bbcdc
    else if 60 ≦ i ≦ 79
        f := b⊕c⊕d; k := 0xca62c1d6
    temp := (a << 5) + f + e + k + w[i]
    e := d; d := c; c := b << 30; b := a; a := temp
```

(5) 中间变量赋值:$\{h_0,h_1,h_2,h_3,h_4\}=\{h_0+a,h_1+b,h_2+c,h_3+d,h_4+e\}$。

(6) 若为最后分组,则第(7)步;否则返回第(1)步处理下一分组。

(7) Hash 值为 h_0,h_1,h_2,h_3,h_4 顺序链接而成(160b)。

SHA-2 算法加强了各个字的位元混合程度,使安全强度得到有效提升。以 SHA-256 为例,算法如下(所有变量为 32b 字长,计算均为 mod 2^{32})。

初始化:h_0 :=0x6a09e667;h_1 :=0xbb67ae85;h_2 :=0x3c6ef372;h_3 :=0xa54ff53a;

h_4 :=0x510e527f; h_5 :=0x9b05688c; h_6 :=0x1f83d9ab; h_7 :=0x5be0cd19。

常量赋值 k[0..63] :=

0x428a2f98	0x71374491	0xb5c0fbcf	0xe9b5dba5	0x3956c25b	0x59f111f1	0x923f82a4	0xab1c5ed5
0xd807aa98	0x12835b01	0x243185be	0x550c7dc3	0x72be5d74	0x80deb1fe	0x9bdc06a7	0xc19bf174
0xe49b69c1	0xefbe4786	0x0fc19dc6	0x240ca1cc	0x2de92c6f	0x4a7484aa	0x5cb0a9dc	0x76f988da
0x983e5152	0xa831c66d	0xb00327c8	0xbf597fc7	0xc6e00bf3	0xd5a79147	0x06ca6351	0x14292967
0x27b70a85	0x2e1b2138	0x4d2c6dfc	0x53380d13	0x650a7354	0x766a0abb	0x81c2c92e	0x92722c85
0xa2bfe8a1	0xa81a664b	0xc24b8b70	0xc76c51a3	0xd192e819	0xd6990624	0xf40e3585	0x106aa070
0x19a4c116	0x1e376c08	0x2748774c	0x34b0bcb5	0x391c0cb3	0x4ed8aa4a	0x5b9cca4f	0x682e6ff3
0x748f82ee	0x78a5636f	0x84c87814	0x8cc70208	0x90befffa	0xa4506ceb	0xbef9a3f7	0xc67178f2

(1) 将分组分为 16 个 32b 字 $w[i], i=0,1,\cdots,15$。

(2) 将 $w[i], i=0,1,\cdots,15$ 扩展成 64 个 32b 字(>>循环右移,>>>逻辑右移,⊕异或,+加):

```
for i from 16 to 63
    s0 := (w[i-15] >> 7)⊕(w[i-15] >> 18)⊕(w[i-15] >>> 3)
    s1 := (w[i-2] >> 17)⊕(w[i-2] >> 19)⊕(w[i-2] >>> 10)
    w[i] := w[i-16] + s0 + w[i-7] + s1
```

(3) 变量赋值:$\{a,b,c,d,e,f,g,h\}=\{h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7\}$。

(4) 主处理程序(循环 64 轮次;& 与,| 或,~非):

```
for i from 0 to 63
    s0 := (a >> 2)⊕(a >> 13)⊕(a >> 22)
    maj := (a& b)⊕(a& c)⊕(b& c)
    t2 := s0 + maj; s1 := (e >> 6)⊕(e >> 11)⊕(e >> 25)
    ch := (e& f)⊕((~e) & g); t1 := h + s1 + ch + k[i] + w[i]
    h := g; g := f; f := e; e := d + t1
    d := c; c := b; b := a; a := t1 + t2
```

(5) 中间变量赋值:$\{h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7\}=\{h_0+a,h_1+b,h_2+c,h_3+d,h_4+e,h_5+f,h_6+g,h_7+h\}$。

(6) 若为最后分组,则第(7)步;否则返回第(1)步处理下一分组。

(7) Hash 值为 $h_0,h_1,h_2,h_3,h_4,h_5,h_6,h_7$ 顺序链接而成(256b)。

SHA-224 与 SHA-256 的算法基本相同,除了 $h_0 \sim h_7$ 的初始值不同、SHA-224 输出时截掉 h_7 的值(因此为 224b)。SHA-512 和 SHA-256 的结构相同,但,SHA-512 处理 64b 字,执行 80 次循环,循环移位量不同。SHA-384 和 SHA-512 的不同点为:$h_0 \sim h_7$ 的初始值不同,SHA-384 输出时截掉 h_6 和 h_7 的值。

12.2.3 MAC 算法

消息认证码(Message Authentication Code,MAC)是用于校验(认证)通信或存储信息的机制,提供鉴别信息是否被篡改的功能,起到保障信息完整性的作用。MAC 集成了

MD5、SHA 等 Hash 函数算法，并采用私钥加密，因此，在 RFC 2104 中被称为 HMAC(Keyed-Hashing for Message Authentication)，如 HMAC_MD5、HMAC_SHA1。

设：H 代表所采用的 Hash 算法；K 为密钥；B 是 H 算法所处理的分组大小(以字节数为单位)；K_B 为 K 后添加数值 0 达到长度 B；O_{pad} 为 0x5c 重复 B 次；I_{pad} 为 0x36 重复 B 次。

又设：$\oplus$为按位异或运算，||将左右两个数据串并合在一起(右边的数据拼接在左边数据的尾部)。对于被鉴别的数据 M，HMAC 算法为

$$H((K_B \oplus O_{pad}) \,||\, H((K_B \oplus I_{pad}) \,||\, M))$$

可使用 HMAC 实现通信双方的身份鉴别。例如执行如下“挑战/响应”认证流程。

(1) 客户机向服务器发出一个鉴别请求。

(2) 服务器接到请求后，生成一个随机数，发送给客户机(挑战)。

(3) 客户机将收到的随机数和存储的密钥进行 HMAC_MD5 运算，将结果作为鉴别证据发送给服务器(响应)。

(4) 服务器也使用该随机数和存储在服务器中的该客户机的密钥进行 HMAC_MD5 运算，如果服务器的运算结果与客户机传回的响应相同，则认为客户机是一个合法用户。

HMAC 引入了密钥，类似一种特殊的加密算法，其安全性已经不完全依赖于所使用的 Hash 算法。HMAC 与私钥加密的区别除了单向性还有一次性，即认证只在当时有效，而私钥加密被破解后，以往的加密结果就被解密了。

12.3　数字签名原理

利用 Hash 函数可以为原消息建立消息摘要，是一种防止消息篡改的方法。一旦改变了输入消息中的任何数据，哪怕只有一位，输出的 Hash 值将会发生不可预测的改变。而且，不论原消息长度有多长，其 Hash 值为定长。这种机制就好比为原消息建立了一个独特的印记，类似人类的指纹，因此，消息摘要也被称为原消息的**消息指纹**(Message Finger-Print)。

通过消息指纹技术可以设计**数字签名**(Digital Signature)机制。数字签名基本原理是：以需要保护的信息作为输入消息，通过单向函数算法，获得消息摘要，并把原消息和消息摘要一同传输；接收方采用相同的方法、相同的单向函数进行计算，并比较输出的结果是否与接收到的消息摘要完全一致，由此判别信息是否被改变(故意篡改或由传输误码造成)。

数字时间戳(Digital Time Stamp，DTS)是数字签名的一种变化，增加了日期、时间信息，可用于时间戳签署业务。

数字签名技术在网络与信息系统中有两种典型的应用方式。

(1) 信息保护。附加在原消息(文件、数据等)上的数字签名起到保护信息完整性的作用，防止信息被非法篡改。例如，一个软件可以和它的数字签名一同保存，任何对软件的修改将使软件不可执行，利用这个机制可以防范病毒感染，也能保护软件的知识产权。

(2) 身份验证。对发送者的“身份声明”进行数字签名，用以保护发送方身份信息，达到身份识别和防抵赖的目的。

从单向函数运用的角度看，数字签名方法非常巧妙，也很理想，然而，单纯使用单向函数

的方案存在很大的安全漏洞：中间人攻击(middleman attack)方法可使用相同的单向函数，在篡改或假冒的消息上附加伪造的数字签名，那么，不但无法防范消息篡改，而且被利用为欺骗手段，甚至具有更强的蒙蔽性。

当 Alice 和 Bob 进行通信时，改进的数字签名流程如下。

(1) 消息发送方(签名方)Alice：

① 创建公钥/私钥对 pubK_a/priK_a，将公钥 pubK_a 发送给 Bob；

② 对原消息 m 做 Hash，得到消息摘要 d；

③ 用私钥 priK_a 加密摘要 d，得到数字签名 D；

④ 发送消息和签名$\{m,D\}$。

(2) 消息接收方(验证方)Bob：

① 分离从 Alice 收到的消息和签名$\{m,D\}$；

② 用 Alice 的公钥 pubK_a 解密签名 D，得到消息摘要 d；

③ 对收到的消息 m 做 Hash，得到消息摘要 d'；

④ 比较 d 和 d'，若相同，说明消息 m 没有被篡改。

采用了单向函数＋私钥加密，攻击者由于无法实施必要的非对称密钥加密步骤，要伪造签名就不那么容易。但是，这个流程仍然是不完善的。如果 Alice 将公钥发送给 Bob 的方式不安全，也会遭受中间人攻击，Bob 收到的也许就是被替换的假冒公钥；而且，密钥管理本身就是一项难题，密钥的生成、分发、存储、使用等任何一个环节不够严谨，都会使整个安全体系分崩离析。所以，一个实用化的网络与信息安全系统应充分运用各种加密算法、安全协议、密钥管理技术，建立严密的认证、保密体系，才能真正达到保障信息可信性、完整性、机密性、可溯性的目标。

CHAPTER 13

第13章 网络安全协议

13.1 密钥安全

13.1.1 Diffie-Hellman 算法

信息加密所使用的密钥需要严格保密，对称加密算法的私钥需要让通信双方安全地同时掌握，非对称加密算法的公钥需要可信地传递，这些都需要密钥安全管理机制来保障。如果密钥管理漏洞百出，再完美的加密算法也毫无用处，所谓的保密体系将形同虚设。

密钥安全管理具体包括三个方面的工作目标：密钥生成（或更新）、密钥分发（或传递）、密钥保管（或托管）。

例如，在一个 n 个用户的网络中，若要进行两两会话，每个会话使用一个密钥，就至少需要 $n(n-1)/2 \approx n^2/2$ 个密钥。100 个用户的网络需要管理约 5000 个密钥。若用户数更多，密钥必然泛滥成灾。因此，必须依靠密钥管理系统进行自动的、安全的、高效的密钥生成和分发工作。

密钥管理以数学算法为基础，通过建立密钥管理系统（如 PKI），为网络中相互通信的计算机服务。

Diffie-Hellman 算法是一种构思非常巧妙的密钥安全管理方法，充分运用了大质数的特性，既实现了对称加密技术的私钥的生成，又同时达到了通信双方安全持有私钥的目的。

DH 算法流程如下。

（1）通信双方 Alice 和 Bob 协商大质数 p 和 q，$1<q<p$，p 和 q 可公开。

（2）Alice 秘密选取大随机数 s，计算 $X=q^s \pmod p$。

（3）Bob 秘密选取大随机数 t，计算 $Y=q^t \pmod p$。

（4）Alice 和 Bob 交换 X 和 Y，并分别计算。

$$\text{Alice：} K_A = Y^s \pmod p\text{；Bob：} K_B = X^t \pmod p$$

显然成立：$K_A=K_B=q^{st} \pmod p$。则 K_A 和 K_B 即分别成为 Alice 和 Bob 握

有的私钥。

DH算法中,虽然 p、q 和 X、Y 是公开的,但由于大随机数 s 和 t 是保密的,攻击者难以据此推算 K_A 和 K_B,而合法通信双方则安全掌握了私钥。

运用非对称密钥ECC加密技术,可设计与DH方法有异曲同工之妙的密钥安全管理算法。

(1) Alice和Bob协商有限域 $GF(2^k)$ 上的椭圆曲线 E,基点 $P\in E(GF(2^k))$,n 为 P 的阶。

(2) Alice随机选取 x,$0\leqslant x\leqslant n$;Alice发送 $k_A=xP$。

(3) Bob随机选取 y,$0\leqslant y\leqslant n$;Bob发送 $k_B=yP$。

(4) Alice计算:$k_{AB}=yK_A$;Bob计算:$k_{BA}=xK_B$。

由于 $k_{AB}=k_{BA}=xyP$,则Alice和Bob成功拥有了相同的私钥。

13.1.2 X.509数字证书

ITU-T X.509标准第1版发布于1988年7月3日,作为X.500的LDAP目录服务标准的一部分,用于单点登录(Single Sign-On,SSO)和授权管理基础设施(Privilege Management Infrastructure,PMI)。X.509定义了(但不限于)公钥证书、证书吊销清单、属性证书和证书路径验证算法等技术标准。X.509第2版(1993年)引入了主体和签发人唯一标识符,以解决主体及签发人名称在一段时间后可能重复使用的问题。1996年发布的X.509第3版支持扩展功能,任何人均可定义扩展并将其纳入证书中。常用的扩展包括:密钥用途(Key Usage)指出密钥用于特殊目的,例如只签;别名(Alternative Names)允许其他标识与公钥证书关联,例如DNS名、电子邮件地址、IP地址等。

数字证书(Digital Certificate)又称公钥证书,颁发者和签署者是合法的、可信的第三方证书颁发机构。数字证书实际上是一个用数字方式签名的真实性得到保证的声明(或名片),其格式符合X.509/RFC 5280(2008年5月)标准。RFC 1442定义了数字证书的信任链,RFC 2459是证书的配置文件,证书数据采用ASN.1(X.690)编码。一张数字证书主要包含以下内容:

(1) 证书的版本信息(V3)。

(2) 证书授予的主体(即证书使用者)名称和标识(X.500格式),如用户名称(例如姓名、公司名称)、目录名、电子邮件地址、域名等,由通用名(Common Name,CN)、组织单位(Organization Unit,OU)、组织(Organization,O)和国家(Country,C)的组合来唯一指代。

(3) 证书的唯一序列号。如果证书被撤销,其序列号将进入证书撤销清单(Certificate Revocation List,CRL)中。

(4) 证书所使用的签名算法(如MD5、SHA等)。

(5) 证书的颁发机构(即CA)名称。

(6) 证书的有效期(UTC时间格式)。

(7) 证书主体的公钥算法和公钥。

(8) 证书的MD5数字摘要(由CA私钥签名)等。

用户通过申请自己的数字证书,可获得CA的公钥、验证证书的公钥、用户自己的私钥(指公钥算法的私钥);用户获得数字证书后,可向CA在线申请并获取经过CA签名认定的

通信对方的公钥。使用这些密钥配合相应的加密算法和操作流程即可完成身份认证、信息加密、密钥分发等重要应用。

数字证书可以存放在计算机或移动介质中，还可以采用 USB-KEY 硬件形式。

如图 13.1 所示，用户（主体）的数字证书由证书内容和数字签名两部分组成，证书内容包含了持有数字证书的主体的身份信息、公钥和认证机构的名称等信息，认证机构用私钥对证书内容进行数字签名，并附加在数字证书中，其他用户只需获得认证机构的公钥，即可验证并获取该数字证书所包含的主体的公钥，从而达到安全获取通信对方公钥的目的。

X.509 数字证书的数据结构如图 13.2 所示。

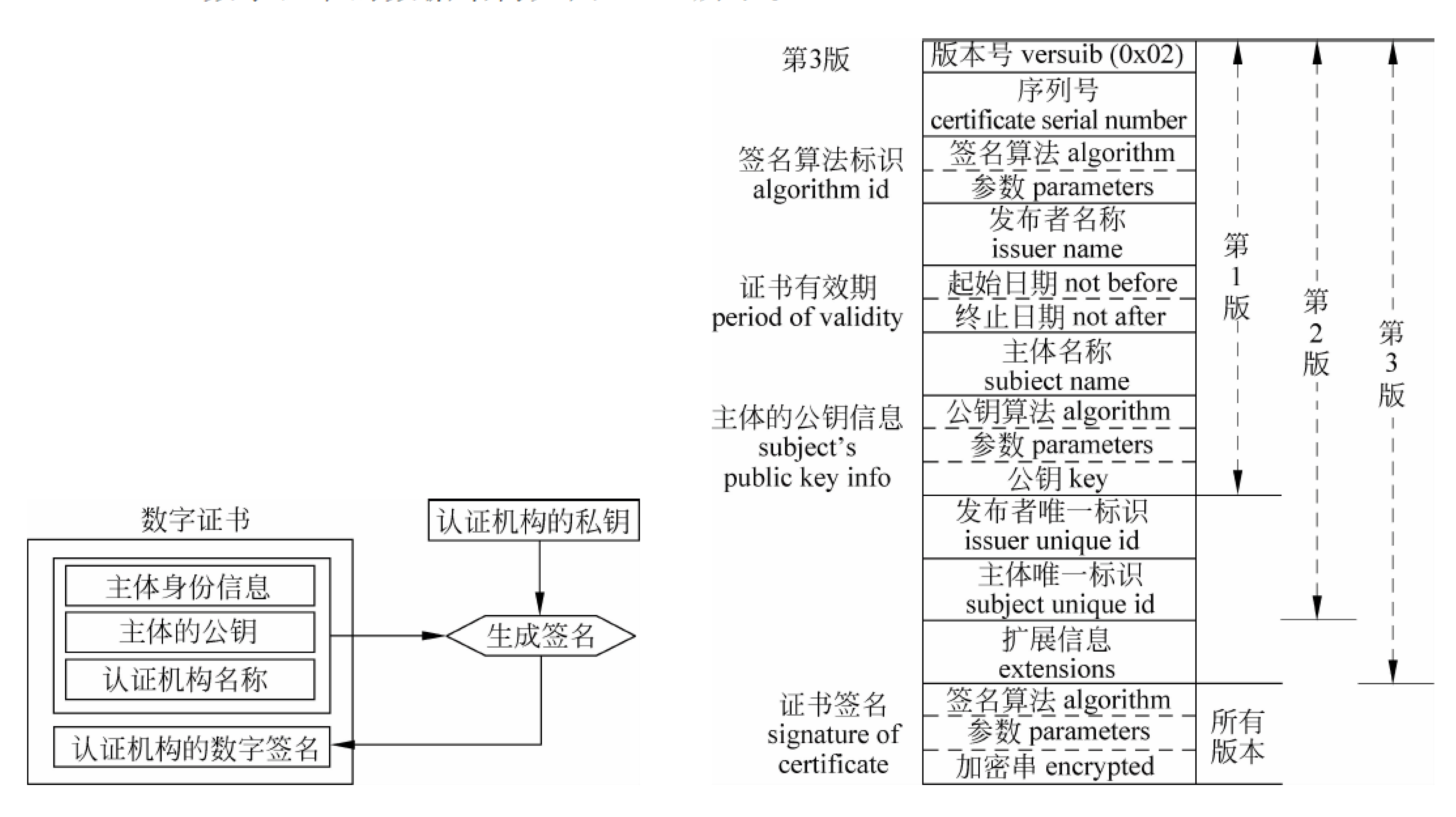


图 13.1　数字证书签名结构　　　　图 13.2　X.509 数字证书数据结构

数字证书可以颁发给法人或自然人，也可颁发给计算机设备；依据应用目标的不同，有身份鉴别证书、电子邮件证书、文件加密证书、安全互连证书、密钥管理证书、版权保护证书和时间戳证书等。

13.1.3　CA

数字证书管理机构（Certificate Authority，CA）是数字证书的颁发者和管理者。CA 被公认为一个采用 X.509 标准化技术的可信的第三方实体机构，是 PKI 体系的核心组成部分。

CA 采用了一种树型结构来建立层次化的信任传递关系，如图 13.3 所示，称为**证书链**（certificate chain）。

（1）根 CA（root CA）具有一个自签名的证书。

（2）从根 CA 开始，依次对下层的从属 CA（subordinate CA）签名。

（3）层次结构中，叶子节点上的 CA 用于对个体进行签名。

（4）对于个体而言，只需信任根 CA，中间 CA 可以不必关心（透明的）。

（5）在每个结点 CA 上，需要保存两种证书：前向证书（forward certificate）是其他 CA

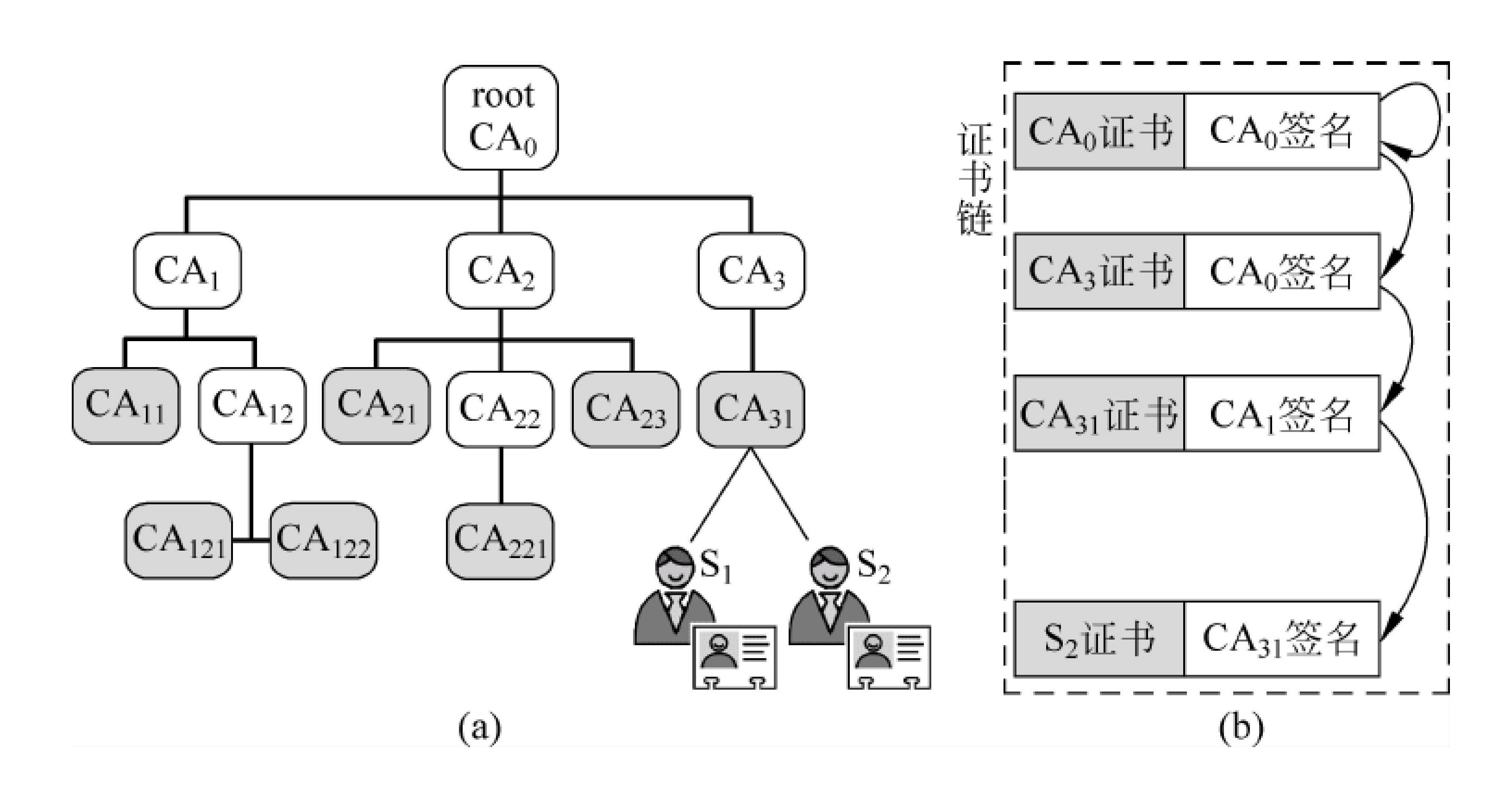


图 13.3　CA 证书链和签名关系

颁发给自己的证书;反向证书(reverse certificate)是 CA 颁发给其他 CA 的证书。

其中,根 CA 是一种特殊的 CA(全球有很多根 CA,相互建立信任关系),受到无条件的信任,位于证书层次结构的最高层,必须对 CA 自己的证书签名,因为在证书层次结构中再也没有更高的认证机构了,所以又称为自签名 CA。另一种为从属 CA,证书中的公钥和用于核实证书的 CA 公钥是不同的。一个 CA 向另一个 CA 颁发证书的过程称为交叉认证,具体有单向交叉认证、双向交叉认证两种情况,并可以进一步细分为域内交叉认证(同一个层次结构内部)和域间交叉认证(不同的层次结构之间)。

CA 层次结构是非常重要的。因为 Internet 不可能只由少数几个 CA 向所有用户提供服务,而是需要许多 CA(通常是本地的、本行业的)同时向用户提供服务。这样,CA 间的信任贯通机制十分必要。例如,计算机操作系统和浏览器中内置了全球可信任的根 CA 证书,当用户获得一张数字证书时,只要对该证书签名的 CA 位于证书链上,则通过信任传递,用户即可验证并信任证书。

例如,Alice 希望在 CA 的帮助下,可信地获取 Bob 的公钥,以便进行下一步的密钥管理和保密通信,则执行如下流程。

(1) Alice 安全地取得 CA 的公钥(Alice 可以在申请并获得自己的数字证书时可靠获得 CA 的公钥)。

(2) Alice 向 CA 请求 Bob 的数字证书。

(3) CA 向 Alice 发送 Bob 的数字证书(其中带有 CA 的私钥签名)。

(4) Alice 接收来自 CA 的 Bob 的数字证书,并验证 CA 签名。

(5) Alice 从 Bob 的数字证书中可信地获得 Bob 的公钥。

CA 负责建设和维护数字证书管理信息系统,并为用户提供证书申请、发放、托管、验证、撤销等服务。CA 系统中,从提供服务功能的角度还可分为 RA、UA 和 WP 等不同角色。

证书登记机构(Register Authority,RA)分散在各处(如银行、大型企业、ISP 等),与 CA 有机结合,接受客户申请、审批申请,并把证书正式请求发送给 CA。RA 的分布式结构有利于 CA 服务网点的开设,具有较好的扩充性,也极大地提高了认证效率。可以把 RA 看做一级弱化的 CA。

证书发布系统(Web Publisher,WP)是 Internet 上普通用户和 CA 直接交流的界面。对用户来说 WP 相当于一个在线的证书数据库。用户的证书颁发之后,CA 通知用户,然后用户使用浏览器从 WP 下载证书。

用户代理(User Agent,UA)或称为目录用户代理(Directory User Agent,DUA)是采用轻量目录访问协议(Light Directory Access Protocol,LDAP)技术建立的系统,接受用户查询、证书回收和更新请求,并直接向用户返回服务结果。在有大量用户的情况下,可在每个 CA 安全域(通常以地理位置分)部署多个 UA,也可在整个 Root CA 管理的证书域部署 UA。

13.1.4　PKI

公钥基础设施(Public Key Infrastructure,PKI)遵循 X. 509 标准,是生成、管理、存储、分发、撤销数字证书所需要的一整套硬件、软件、人员、策略、过程等,可视为实现 CA 的完整体系结构。PKI 包含一系列公共密钥加密标准(Public-Key Cryptography Standards,PKCS),如:

- PKCS #1——RSA Encryption Standard
- PKCS #3——Diffie-Hellman Key-Agreement Standard
- PKCS #5——Password-Based Encryption Standard
- PKCS #6——Extended-Certificate Syntax Standard
- PKCS #7——Cryptographic Message Syntax Standard
- PKCS #8——Private-Key Information Syntax Standard
- PKCS #9——Selected Attribute Types
- PKCS #10——Certification Request Syntax Standard
- PKCS #11——Cryptographic Token Interface Standard
- PKCS #12——Personal Information Exchange Standard
- PKCS #13——Elliptic Curve Cryptography Standard
- PKCS #15——Cryptographic Token Information Format Standard

PKI 系统具有数字证书管理机构(CA)、数字证书库、密钥备份及恢复系统、证书撤销清单(CRL)系统、应用程序接口(API)等基本构件,建立 PKI 也将围绕着这五大系统来进行。

PKI 提供完善的信息安全所需的密钥管理和服务环境。以 Alice 与 Bob 进行安全电子邮件通信为例,可执行如下操作流程。假定 Alice 和 Bob 事先都已经拥有数字证书。

(1) Alice 从 CA 安全获得 Bob 的公钥(见 13.1.3 节)。

(2) Alice 创建一个对称密钥加密的会话密钥。

(3) Alice 使用该会话密钥加密邮件明文。

(4) Alice 使用 Bob 的公钥将该会话密钥加密。

(5) Alice 将邮件密文和加密密钥一起发送(根据电子邮件应用特点)。

(6) Bob 使用自己的非对称加密算法私钥来解密会话密钥。

(7) Bob 使用会话密钥解密邮件密文,得到明文。

思考:为何当需要发送保密邮件时,通信双方都需要拥有数字证书,而发送数字签名邮件时,只需发送方拥有数字证书?

13.2 安全认证

13.2.1 PAP

口令鉴别协议(Password Authentication Protocol,PAP)是PPP中采用的一种安全协议,在初始链路建立的基础上,通过二次握手机制提供一种对等结点间身份认证的简单方法。

PPP在LCP协商阶段配置的鉴权协议选择参数为<type> <length> <authen-protocol>=<03 | 04 | c0 23>,表示采用PAP。PAP的报文格式如图13.4所示。

PAP报文类型有:1=鉴别请求(Authenticate Request),2=鉴别确认(Authenticate Ack),3=鉴别否认(Authenticate Nak)。请求及其确认(否认)报文应具有相同的ID号。

PAP非常简单,即用户终端发送带有用户名和口令的鉴别请求报文给服务器,服务器经过口令认证,返回鉴别确认或否认的报文。

显然,PAP存在比较验证的安全性缺陷,因为用户名和口令均以文本字符格式(明文)在线路上进行传输,对于通信窃听、重放攻击、反复尝试等没有任何防范能力。

0	类型
1	ID
2 3	总长度
4	用户名长度
·	用户名字符串 (明文)
·	口令长度
·	口令字符串 (明文)

图13.4 PAP报文格式

13.2.2 CHAP

挑战握手鉴别协议(Challenge-Handshake Authentication Protocol,CHAP)是PPP的安全认证协议之一,在RFC 1994中定义。

PPP在LCP协商阶段配置的鉴权协议选择参数为<type> <length> <authen-protocol> <algorithm>= <03 | 05 | c2 23 | 05>,表示选用CHAP和MD5单向函数算法。

CHAP的报文格式为<code 8bit> <identifier 8bit> <length 16bit> <data>。其中code表示的报文类型有1=Challenge,2=Response,3=Success,4=Failure。请求和响应报文均具有相同的identifier编码(用以配对),而每个Challenge报文的identifier都应与前一个发送的不同。

挑战(challenge)是一种早期军队狭路相逢时(尤其在黑暗中)辨别敌我的方法。如果一方说"一匹马"作为挑战词,另一方应回答事先秘密约定好的绝密口令列表第一项"土豆";如果挑战为"八达岭",回答则为第八项"老虎"。回答错误者绝非友军。

为保障安全性,CHAP通信双方采用秘密数(secret)单向加密的检验机制。但secret并不在网络上传输,而是由双方保存同样的secret数据表,这样双方都可以验证。网上传输的是用于检索secret的挑战值和secret的单向函数加密值。要求挑战值和secret每次都不重复(一般可为大随机数),并且从挑战值无法推算出secret。

对于Challenge和Response报文,data字段采用如下结构:<value-size> <value> <name>。Challenge报文的value中存放挑战值,长度由value-size决定,可由一个或多个字节组成,最高字节先发送;Response报文的value中存放由挑战值检索到的secret的

Hash 结果（如果采用 MD5 算法，则为 16B）。name 字段的长度由 CHAP 报文总长度 length 决定，存放一个或多个字节的报文发送方主机名称或账号字符串，在多个用户接入的情况下，可用于相互区分。

对于 Success 和 Failure 报文，分别表示鉴别成功和失败，data 字段存放文本型消息，长度由 length 值决定，用于解释鉴别的结果（可供显示）。

如图 13.5 所示，CHAP 通过三次握手机制周期性地鉴别通信对端的身份，可在初始链路建立时进行鉴别，并在链路建立之后重复进行。通过递增改变的标识符和可变的挑战值，CHAP 可有效防止来自非法用户端的重放攻击。虽然该流程实现的是单向认证，但任意一方均可按此流程发起对对方身份的鉴别。

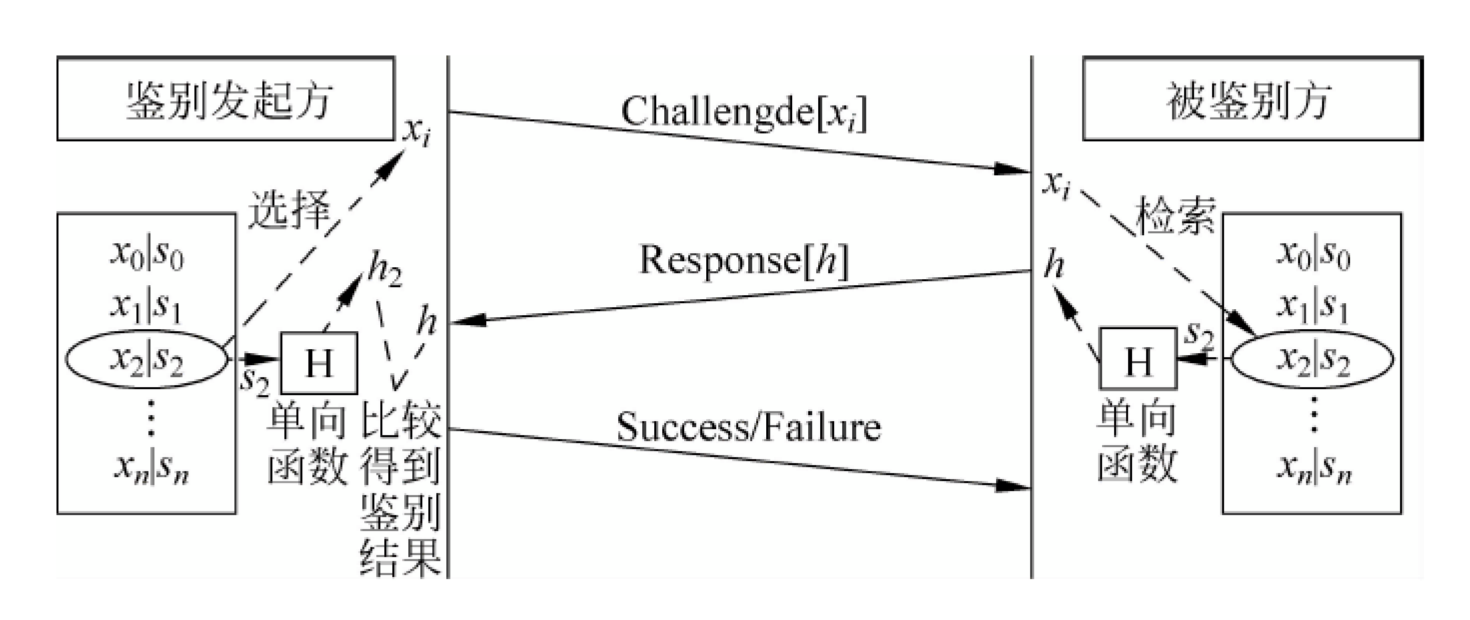


图 13.5　CHAP 鉴别三次握手机制

CHAP 可用于用户登录的安全认证。由服务器为鉴别发起方，发送以随机数（每次不同）作为挑战值的 Challenge 报文；用户方回复将用户名作为 name 字段内容、将口令和挑战值混合后计算 Hash 值的 Response 报文；服务器可根据用户名检索口令表并按同样方式计算 Hash 值，与收到的 Response 报文中的值比较，即可鉴别用户合法性。

13.2.3　RADIUS 协议

远程拨号用户认证服务（Remote Authentication Dial-In User Service，RADIUS）是应用非常广泛的 AAA 协议，即提供鉴别（authentication）、授权（authorization）及记账（accounting）三类服务。1997 年发布了 RFC 2058 标准，随后是 RFC 2138，最新版本的 RFC 2865/2866 发布于 2000 年 6 月。

RADIUS 是一种 C/S 结构的协议，其客户机并非通常的网络登录用户计算机，而是网络接入服务器（Network Access Server，NAS）、通信服务器（Access Concentrator，AC）、应用服务器、Web 服务器、数据库服务器等，还包括运行 RADIUS 客户机软件的其他网络设备。RADIUS 协议认证机制灵活，可以为 PAP、CHAP 或者操作系统登录等多种方式提供认证服务。RADIUS 也是一种可扩展的协议，其全部工作都是基于{attribute，length，value}向量进行的，并支持扩充不同设备专用的属性。IEEE 提出的用于无线网络接入认证的 802.1x 标准同样采用 RADIUS 协议。

RADIUS 协议采用 UDP 传输，端口 1812 用于认证服务，端口 1813 用于记账服务。RADIUS 报文格式如图 13.6 所示。

Code 字段指出 RADIUS 报文类型：1=服务请求（Access Request）；2=服务接受（Access Accept）；3=访问拒绝（Access Reject）；4=记账请求（Accounting Request）；5=记账响应

(Accounting Response);11=访问挑战(Access Challenge)。

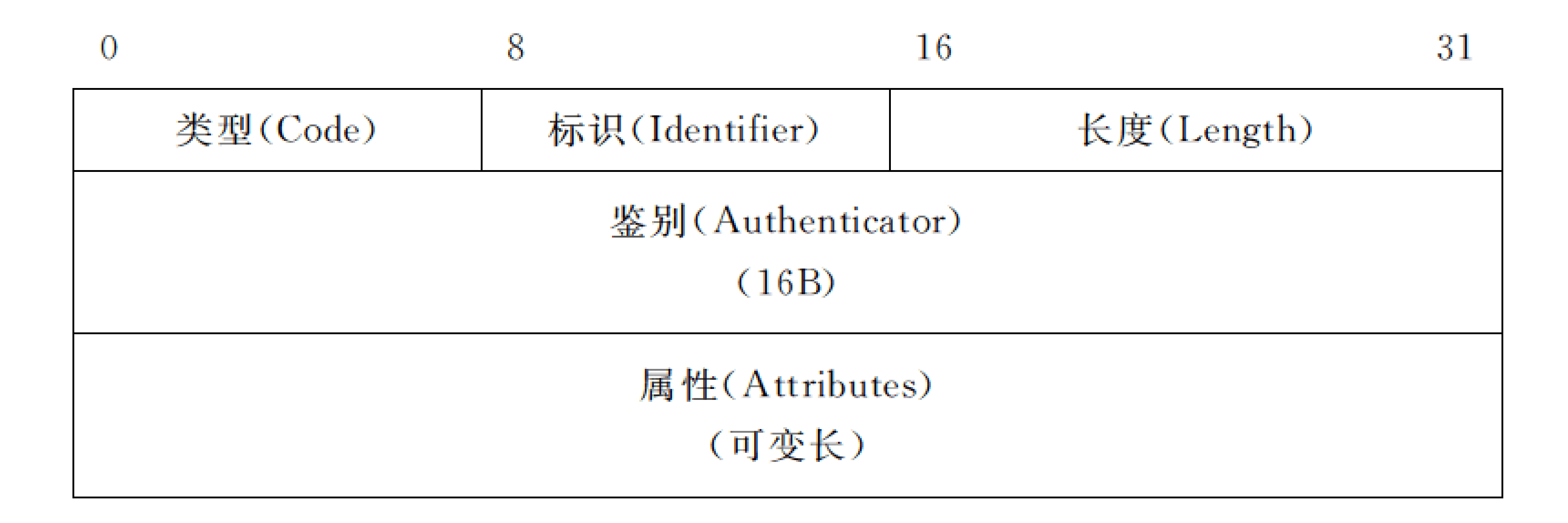


图 13.6　RADIUS 报文格式

Authenticator 字段为 16B 的鉴别字,用于 RADIUS 客户机和服务器之间判别消息有效性。Access Request 报文中鉴别字的值是 16B 随机数(设为 Random),要求不能被预测并且在一个共享密钥(设为 Secret)的生命期内唯一。其他报文的鉴别字分别定义为:

Access Accept、Access Reject 和 Access Challenge:

A_1 = MD5(Code + ID + Length + Random + Attributes + Secret)

Accounting Request:

A_2 = MD5(Code + ID + Length + 16ZeroOctets + Attributes + Secret)

Accounting Response:

A_3 = MD5(Code + ID + Length + A_2 + Attributes + Secret)

Identifier 字段是用于匹配请求和响应报文的标识符(ID)。

Attributes 字段用于传送服务类型、用户名、口令、IP 地址、端口号、协议类型等信息,格式为<type 8b> <length 8b> <value>。例如:type=1 为用户名,type=2 为用户口令,type=3 为 CHAP 口令等。用户口令不应在网络上传输,可采用如下方法封装。

设共享秘密字(secret)为 S,128b 的伪随机数鉴别字为 RA。将口令字符串分隔为 16B 数据块 p_i,$i=1,2,\cdots$,最后一块以 0 填充。计算:

$$b_1 = \text{MD5}(S+RA),c(1) = p_1 \text{ xor } b_1;$$
$$b_2 = \text{MD5}(S+c(1)),c(2) = p_2 \text{ xor } b_2;$$
$$\cdots$$
$$b_i = \text{MD5}(S+c(i-1)),c(i) = p_i \text{ xor } b_i$$

最后级联 $c(1)+c(2)+\cdots+c(i)$,作为用户口令传送时 value 字段的内容。认证服务器可以按同样算法进行验证。

RADIUS 协议工作流程如图 13.7 所示。

RADIUS 还支持代理和漫游功能。代理服务器负责转发 RADIUS 认证和记账报文;漫游功能则是代理的一个具体实现,使用户可以使用其他非归属地的 RADIUS 服务器进行认证。

思考:为什么不是直接在 NAS 上验证用户而需要使用 RADIUS 协议?

13.2.4　Kerberos 协议

Kerberos 协议是一种计算机网络鉴权协议,用于在非安全网络中对访问者以安全的手段进行身份认证。Kerberos 是 MIT 为该协议开发的一套软件的名称。

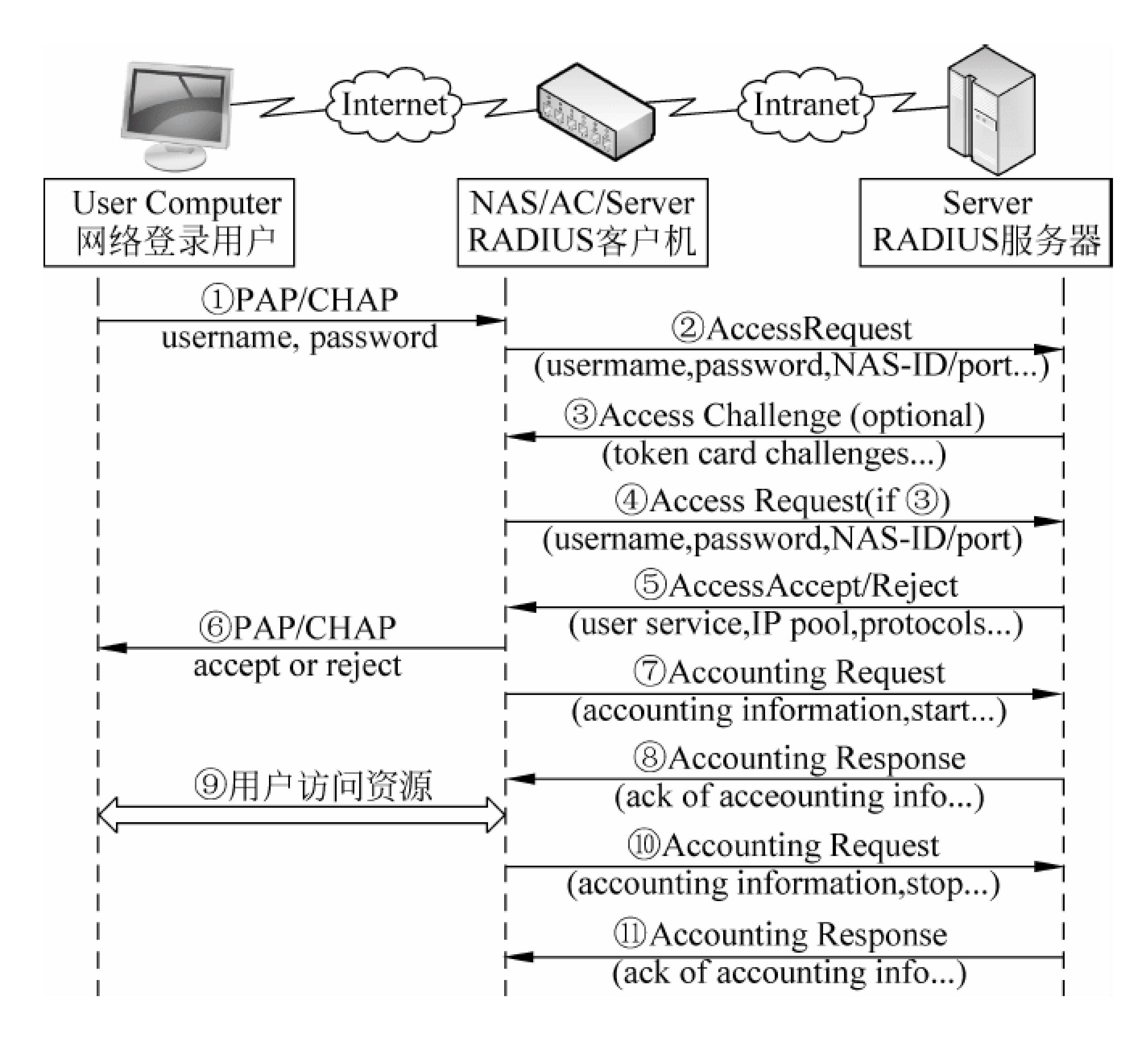


图 13.7　RADIUS 协议工作流程

用于命名协议的 Kerberos 是希腊神话中一条凶猛的三头神犬(又名 Cerberus),是地狱之门的守护者。

最初版本的 Kerberos 在 1980 年末发布,基于 Needhan-Schroeder 算法。版本 5 在 1993 年作为 RFC 1510 颁布,之后在 2005 年被 RFC 4120 取代,相关标准还有 RFC 3961/3962/4121。

Kerberos 系统采用 C/S 结构,支持通信双方的相互认证,可以防止数据窃听、防止重放攻击、保护数据完整性。Kerberos 运用对称密钥加密体制进行密钥管理,其扩展方法也可使用公开密钥加密方法。当有 N 个用户使用该系统时,为确保在任意两方之间进行秘密对话,系统维护与每个用户的共享密钥,所需的最少会话密钥数为 N 个。

Kerberos 由两个独立的逻辑部分组成:认证服务器(Authentication Server,AS)和票据授权服务器(Ticket Granting Server,TGS),共同形成可信赖的第三方,称为**密钥分发中心**(Key Distribution Center,KDC)。Kerberos 的工作基于用来证明用户身份的票据(ticket)。用票据授权的票据(也称票据的票据)简称 TGT(Ticket Granting Ticket)。为区别于 KDC 服务器,把提供应用服务的服务器称为 SS(Service Server)。

KDC 拥有一个密钥数据库;每个网络实体,无论是客户终端还是服务器,共享一套只有自身和 KDC 知道的密钥,该密钥用于证明实体的身份。对于两个实体间的通信,KDC 产生一个会话密钥,用来加密交互信息。

如图 13.8 所示,用户终端(客户机)在获得 SS 服务前,应首先和 KDC 间使用对称密钥加密(如 DES、AES)的 Kerberos 协议进行认证。

认证协议的执行过程如下。

(1) 用户使用客户机登录。

① 用户输入用户名 U(用户 ID)和口令 P。

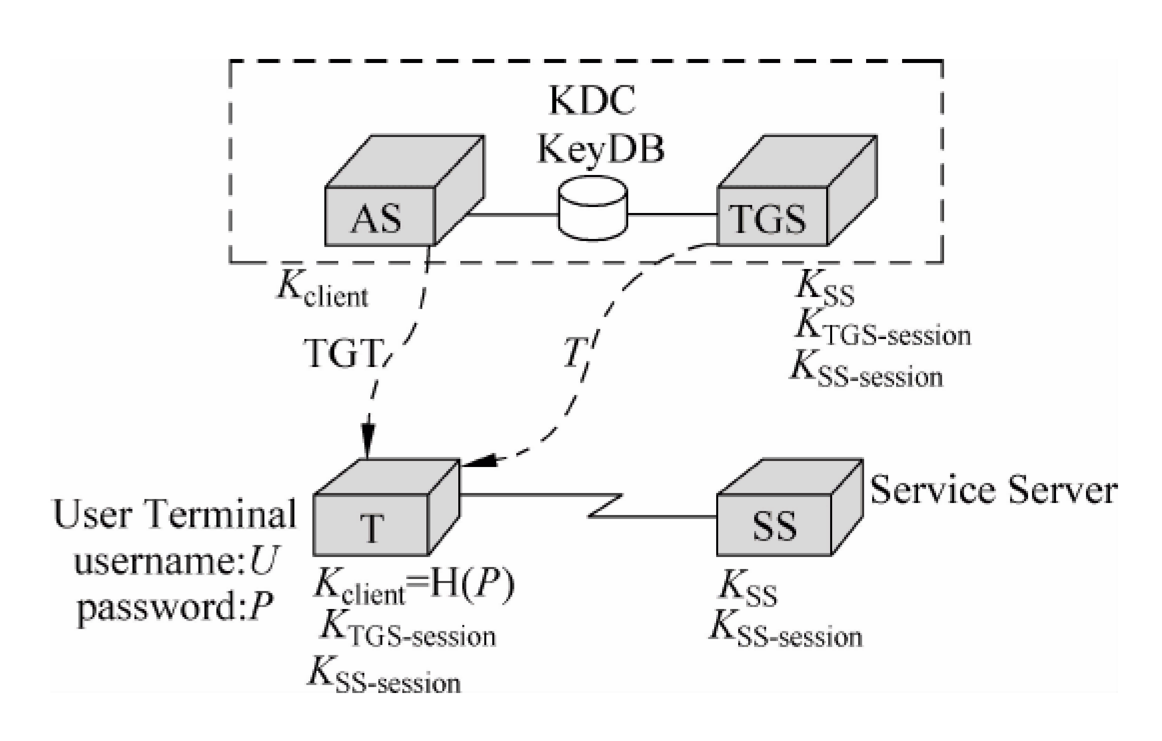


图 13.8　KDC 与客户机和服务器认证关系示意

② 客户机程序计算 $K_{client}=\text{Hash}(P)$,作为客户机的用户密钥。受信任的 AS 通过安全的途径事先已获取了相同的密钥。

(2) 客户机认证(客户机从 AS 获取 TGT)。

① 客户机向 AS 发送一条包含用户 ID 的明文信息,表明用户 U 请求服务。

② AS 检查用户 ID 有效性,并返回两条消息。

a. 消息 A:用 K_{client}加密的 $K_{TGS\text{-}session}$(客户机-TGS 会话密钥,用于将来客户机与 TGS 的会话)。

b. 消息 B:用 K_{TGS}(TGS 密钥)加密的 TGT(TGT 包括 $K_{TGS\text{-}session}$、U、用户网址、TGT 有效期)。

③ 客户机 K_{client}解密消息 A 内容,得到 $K_{TGS\text{-}session}$。注意:客户机不能解密消息 B,因为是用 K_{TGS}加密的。

(3) 服务授权(客户机从 TGS 获取票据 T)。

① 客户机向 TGS 发送以下两条消息。

a. 消息 c:收到的消息 B 和想获取的服务的服务 ID。

b. 消息 d:用 $K_{TGS\text{-}session}$加密的认证符(包括 U 和时间戳)。

② TGS 用 K_{TGS}解密消息 c 中的消息 B 得到 TGT,从而得到 AS 提供的 $K_{TGS\text{-}session}$;然后用 $K_{TGS\text{-}session}$解密消息 d 得到 U;而后向客户机发送两条消息。

a. 消息 E:用 K_{SS}(服务器 SS 密钥)加密的 T(T 包括客户机-SS 会话密钥 $K_{SS\text{-}session}$、U、用户网址、T 有效期)。

b. 消息 F:用 $K_{TGS\text{-}session}$加密后的 $K_{SS\text{-}session}$。

③ 客户机用 $K_{TGS\text{-}session}$解密消息 F,得到 $K_{SS\text{-}session}$。注意:客户机不能解密消息 E,因为是用 K_{SS}加密的。

(4) 服务请求(客户机从服务器 SS 获取服务)。

① 客户机向 SS 发出两条消息。

a. 消息 e:消息 E。

b. 消息 g:用 $K_{SS\text{-}session}$加密后的新认证符(包括 U 和时间戳)。

② SS 用 K_{SS}解密消息 E(即 e)得到 T,从而得到 TGS 提供的 $K_{SS\text{-}session}$。用 $K_{SS\text{-}session}$解密消息 g 得到 U,而后向客户机返回一条确认消息(意为:确证身份真实,乐于提供服务)。

消息 H 是用 $K_{SS\text{-}session}$加密后的新时间戳(客户机发送的时间戳加 1)。

③ 客户机用 $K_{SS\text{-}session}$解密消息 H,得到新时间戳。

④ 客户机发现时间戳在消息 H 中被正确地更新，表明客户机可以信赖服务器(SS)，因此向 SS 发送服务请求。

⑤ 服务器(SS)响应并提供服务。

13.3　TCP/IP 安全

13.3.1　PPTP

点对点隧道协议(Point-to-Point Tunneling Protocol，PPTP)是一种支持多协议虚拟专用网络(Virtual Private Network，VPN)的技术，由 RFC 2637 定义，工作在第二层，协议规范本身并未描述数据加密或身份验证的特性。

PPTP 工作在点对点链路上，可实现网络终端通过 ISP 拨号上网并安全接入企业内部网络。PPTP 允许对 IP、IPX 或 NetBEUI 数据流进行加密，然后封装在 IP 报文中，通过 Intranet 或 Internet 来相互通信。PPTP 在远程客户端和内部网络的服务器之间运行，网络互连设备不需要介入，因此，PPTP 构成了穿越 IP 网络的安全隧道。这种通过 IP 报文在 Internet 上透明传输信息的方式被称为**通用路由封装**(Generic Route Encapsulation，GRE)技术(RFC 2784)。

GRE 可在任意一种网络层协议报文中封装另一种协议报文，报头格式为：

```
c;                1b; c = 1:校验字有效
reserved0;        12b; 保留
version;          3b; 版本号,现为 0
protocol - type;  16b; 协议类型(RFC 1700 之 Ether Types),IP:0x0800
check - sum;      16b; 校验字(可选字段)
reserved1;        16b; 保留(可选字段)
```

PPTP 由两个并列的元素组成：控制连接工作在 TCP 上(端口号为 1723)，在通信双方间操作；控制连接建立后，IP 隧道操作运行在相同的通信双方之间，传输 GRE 封装的 PPP 报文(IP 的协议标识为 47)，实现用户会话。

PPTP 的报文格式如图 13.9 所示(不同的 PDU 格式上有所不同)。PPTP 消息类型有两种：1 为控制消息，2 为管理消息。

←4B(32b)→

报文总长度(length)	PPTP 消息类型(PPTP_type)
Magic Cookie＝0x1A2B3C4D	
控制消息类型(control_type)	保留(0)
协议版本(version)或原因码	保留(0)或原因码
帧传输类型(Framing Capabilities)(1:异步；2:同步)	
承载信道类型(Bearer Capabilities)(1:模拟；2:数字)	
最大 PPP 会话数(max. chnl.)	固件版本(firmware revision)
主机名称(Host Name) (64B；用 0 填充)	
供应商字串(Vendor String) (64B；用 0 填充)	

图 13.9　PPTP 报文格式

(1) 控制消息类型包括：1＝启动控制连接请求，2＝启动控制连接回复，3＝停止控制连接请求，4＝停止控制连接回复，5＝回显请求，6＝回显回复。

(2) 管理消息类型包括：1＝出呼叫请求，2＝出呼叫回复，3＝入呼叫请求，4＝入呼叫回复，5＝入呼叫接通，6＝呼叫清除请求，7＝呼叫断开报告，8＝广域网错误报告。

如图 13.10 所示为 PPTP 控制连接操作发送、接收的报文，以及发起方和接收方的状态转换关系。

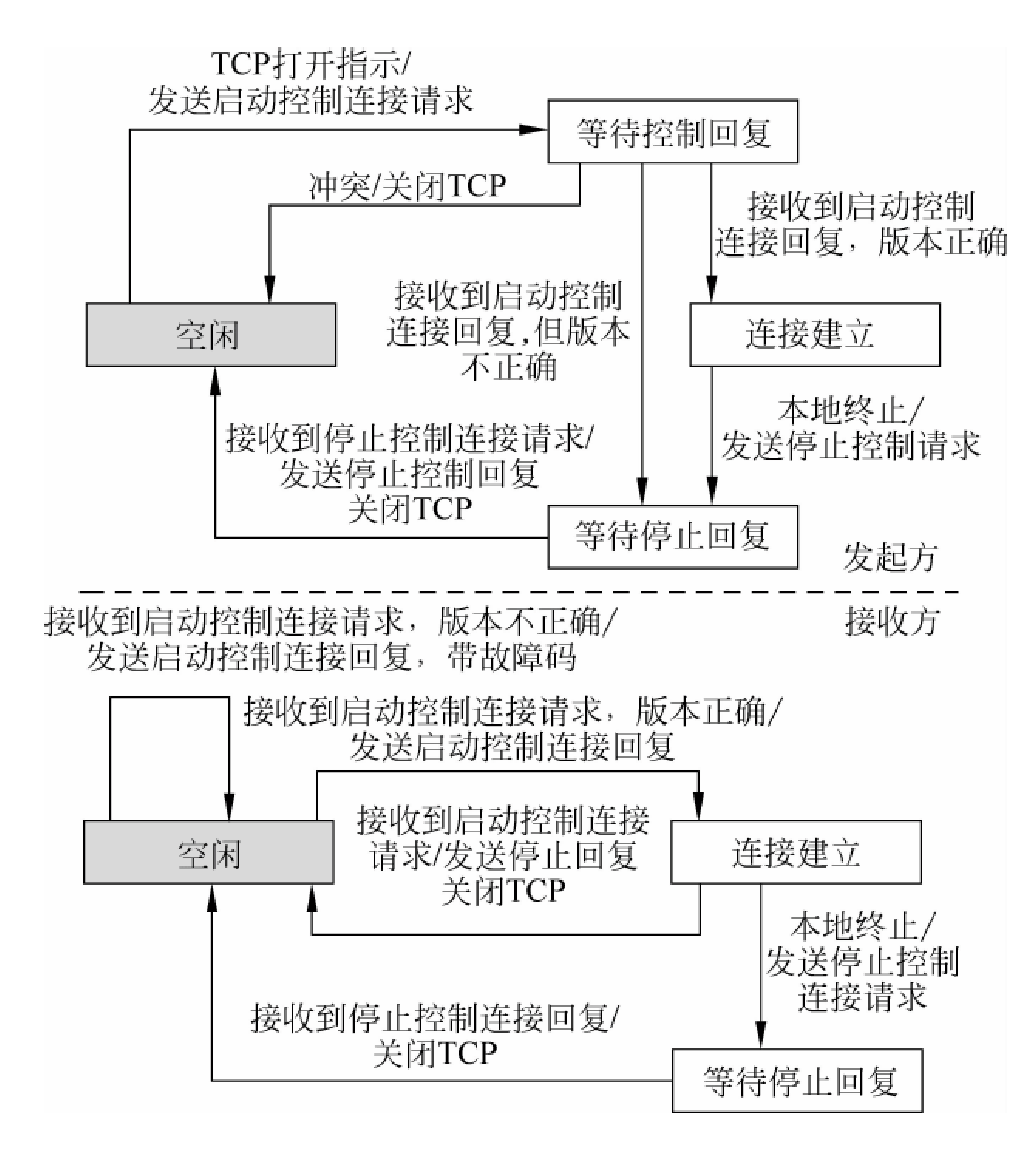


图 13.10　PPTP 控制连接发起方和接收方状态机

13.3.2　L2TP

第二层隧道协议(Layer 2 Tunneling Protocol，L2TP)与 PPTP 的功能类似，且都属于数据链路，使用 PPP 对数据进行封装，然后添加附加报头用于数据在互联网上的传输，用于 VPN 安全互连，但是存在以下差异。

(1) PPTP 要求基于 IP 网络；L2TP 只要求隧道为面向分组的点对点连接，可以在 IP、FR(PVC 方式)、X.25 或 ATM 网络上使用。

(2) PPTP 只能在两端点间建立单一隧道；L2TP 支持在两端点间使用多隧道，用户可以针对不同的服务质量创建不同的隧道。

(3) L2TP 可以提供包头压缩。当压缩包头时，系统开销(overhead)占用 4B，而 PPTP 下要占用 6B。

(4) PPTP 不支持隧道验证，L2TP 则可以提供隧道验证。但是，当 L2TP 或 PPTP 与 IPSec 共同使用时，可以由 IPSec 提供隧道验证，并不需要在第二层协议上验证。

在 IP 网络中，L2TP 使用 UDP(端口号 1701)传输报文，从这个意义上说，L2TP 又像是

一个会话层协议。L2TP 于 1999 年 8 月由 RFC 2661 定义(v2),RFC 3817 定义了用于 PPPoE 的 L2TP 主动发现中继协议,2005 年发布了 RFC 3931 定义的 v3 版本。L2TP 协议栈的结构如图 13.11 所示。

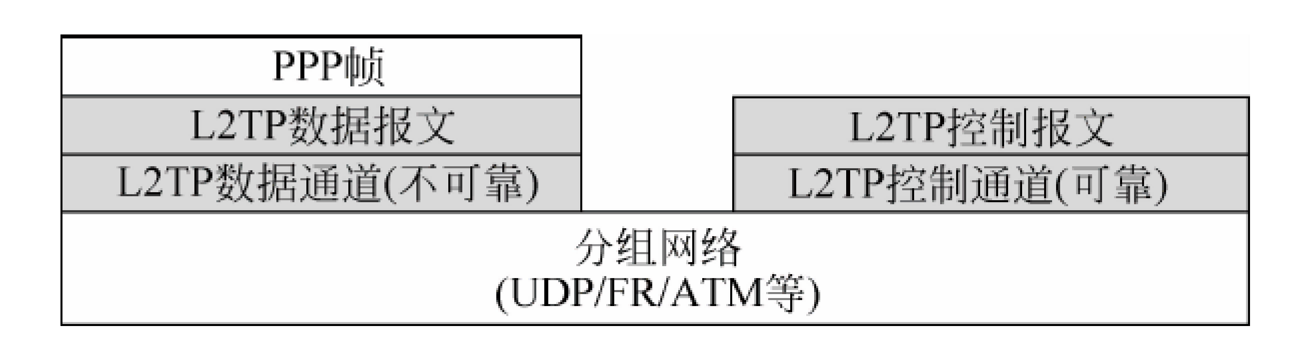


图 13.11　L2TP 协议栈

L2TP 报头格式如图 13.12 所示。

0								8					16　　　31
T	L	x	x	S	x	O	P	x	x	x	x	ver(2)	总长度(Length)(可选)
隧道标识(Tunnel ID)													会话标识(Session ID)
发送序号(Ns)(可选)													接收序号(Nr)(可选)
偏移量大小(Offset Size)(可选)													偏移量填充(Offset pad)(可选)

图 13.12　L2TP 报头格式

T 比特表示报文类型,0 为数据报文,1 为控制报文。L 比特为 1 表示长度字段有效。保留的 x 比特均置 0。S 比特为 1 表示 Ns 和 Nr 序号有效。O 比特为 1 表示偏移量大小字段有效,在控制报文中 O 比特应置 0。P 比特置 1 表示优先处理报文。隧道标识用于表示控制连接,而会话标识指示隧道内的一个会话。

L2TP 控制报文的类型与 PPTP 类似。

13.3.3　IPSec

安全 IP(Secure IP,IPSec)是 IP 的一个子层(RFC 2401～2411),可以看做 IP 的安全补丁,或称为 3.5 层协议,用于 VPN 安全互连。

IPSec 分为两个部分。

(1) 鉴别报头(Authentication Header,AH)。AH 为 IP 报文提供数据完整性保障和源站鉴别功能,但不提供数据保密功能。AH 报头由目的站进行解析和处理,例如可通过数字摘要检查报文完整性。AH 报文数据结构如图 13.13 所示。

(2) 安全封装(Encapsulating Security Payload,ESP)。ESP 提供数据加密、源站鉴别和数据完整性保障功能。ESP 将需要保密的用户数据和 ESP 尾部一起进行加密后,再封装到一个新的 IP 报文中(如图 13.14 所示)。AH 中的 next-header 字段被移到 ESP 尾部,防止攻击者了解 TCP/UDP 协议类型。

AH 或 ESP 首先建立安全关联(Security Association,SA)。SA 是一个单向的逻辑连接,包括标识符(即 SPI 值)、目的 IP 地址和安全协议类型(AH/ESP)3 个要素。SA 将决定安全策略,指定加密算法和认证算法。

AH 和 ESP 均有两种协议封装模式(如图 13.15 所示)。

(1) 传输模式(transport mode):不改变 IP 报头,插入一个 AH/ESP。

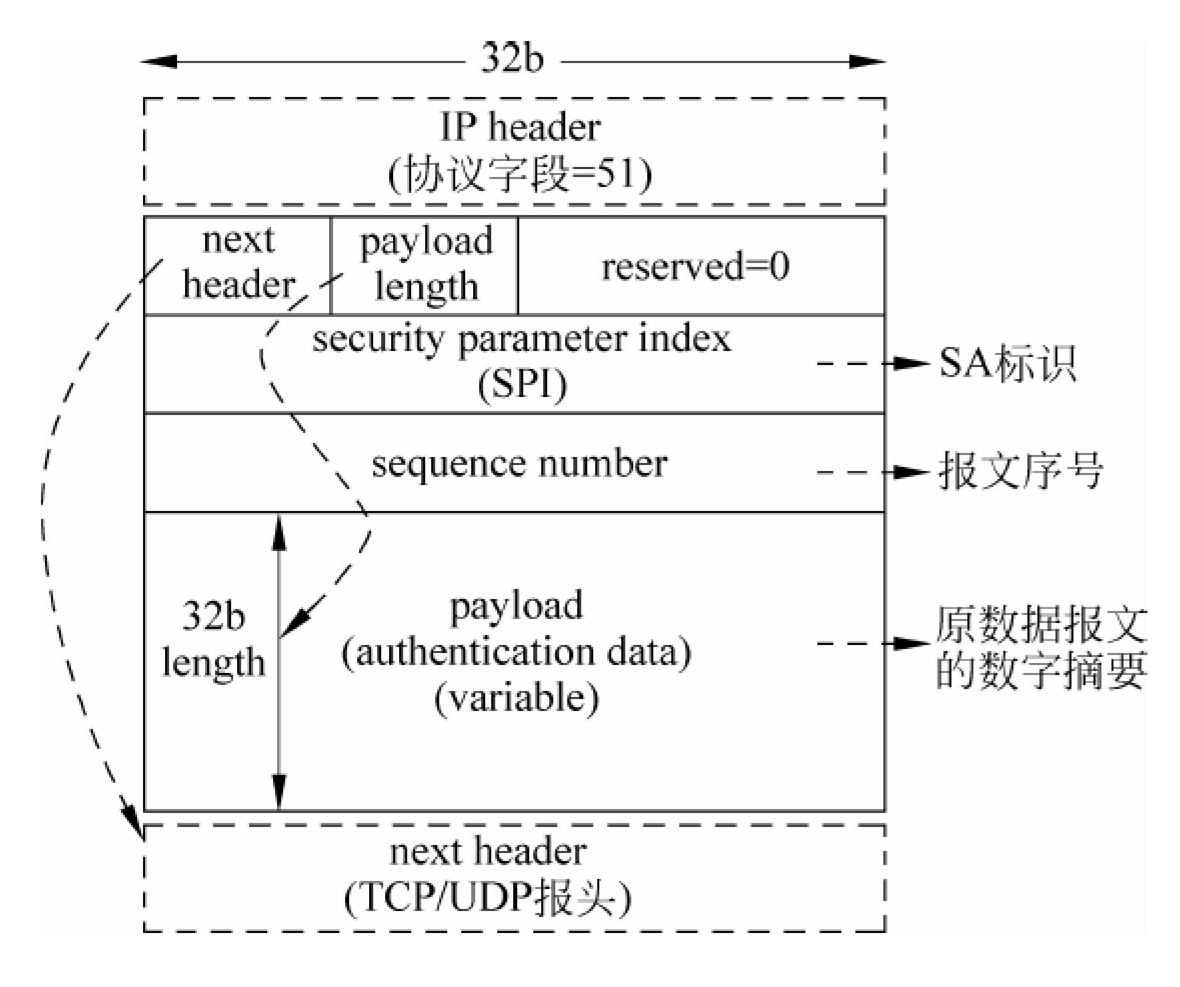


图 13.13　AH 报头格式

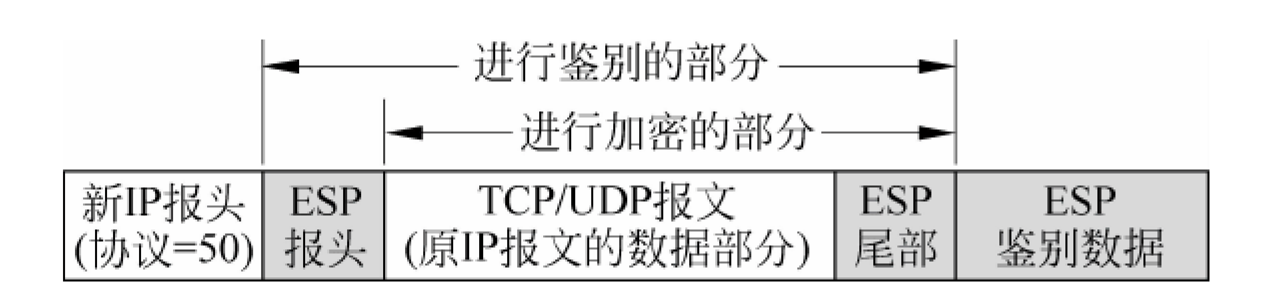


图 13.14　ESP 报头结构

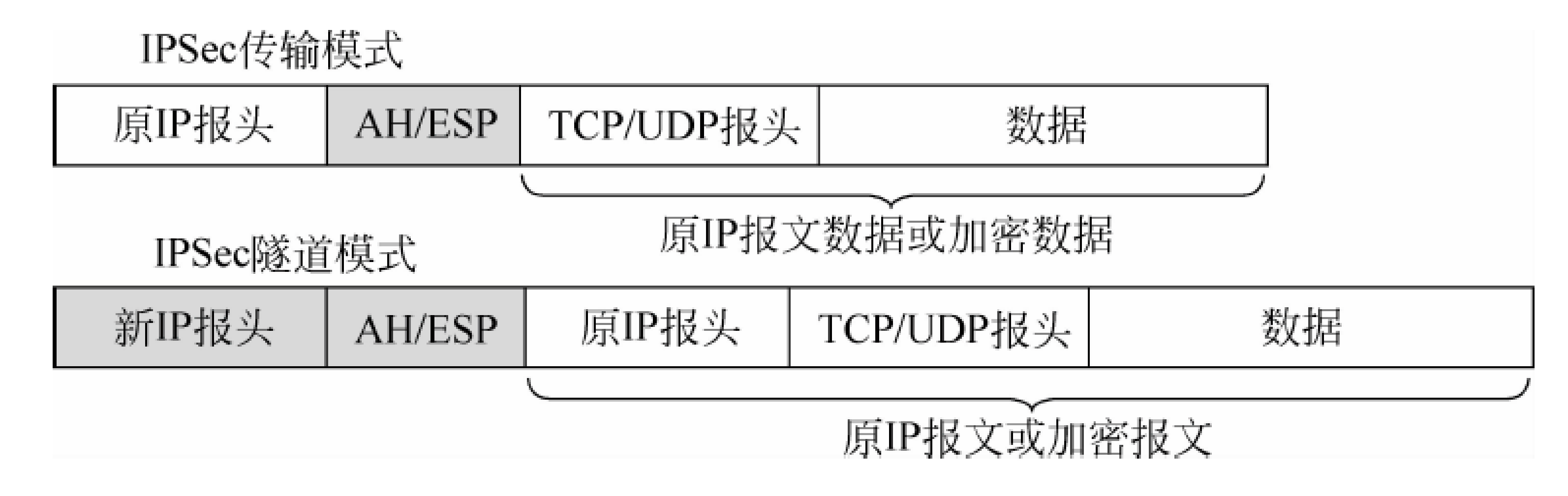


图 13.15　IPSec 协议封装模式

(2) 隧道模式(tunnel mode):生成一个新的 IP 报头,把 AH/ESP 和原来的整个 IP 报文(或加密报文)放到新 IP 报文的载荷数据中。

13.3.4　SSL 协议

安全套接字层(Secure Socket Layer,SSL)协议(RFC 2246)用于在 Internet 上进行保密的、可信的通信,是 TCP 之上的运输层子层(也可视为应用层的子层),IETF 称为 TLS(Transport Layer Security),其最新版本是 RFC 5246(版本 1.2)。SSL 已取代应用层的 SHTTP(安全 HTTP)。

SSL 提供三大网络安全功能。

(1) SSL 服务器鉴别:浏览器采用合法可信的 CA 数字证书及其包含的公钥,通过 SSL 对服务器身份进行 CA 证书鉴别。

(2) 加密 SSL 会话:生成并安全交换会话密钥,会话数据由发送方加密,接收方解密,防止信息泄露。使用 RC4 流式加密算法,会话密钥长度为 40b。

(3) SSL 用户鉴别：服务器也可以采用数字证书鉴别用户的身份。

SSL 协议的优势在于与应用层协议独立无关性。应用层协议(HTTP、FTP、Telnet 等)都能透明地运行于 SSL 协议之上。SSL 协议简要工作方式如下。

(1) 客户端发送 Client-Hello 消息，说明支持的密码算法列表、压缩方法及最高协议版本，并发送稍后使用的随机数。

(2) 客户端收到服务器发送的 Server-Hello 消息，包含服务器从 Client-Hello 消息中选择的连接参数。

(3) 协商完毕连接参数后，客户端与服务器交换数字证书(依靠被选择的公钥系统)。数字证书通常基于 X.509 标准，也可支持以 OpenPGP 标准为基础的证书。

(4) 服务器请求客户端公钥。客户端若有证书可作双向身份认证，若无证书则随机生成公钥。

(5) 客户端与服务器使用随机数通过公钥加密技术安全地协商私钥，或利用 Diffie-Hellman 算法安全生成和分发私钥。所有关键数据的加密均使用该私钥。

数据传输过程中，记录层(record layer)用于封装高层的 HTTP 等协议。记录层数据可以与消息验证码一起被压缩、加密。每个记录层报文都有一个 content-type 字段用以标识上层的协议。

SSL 的最常见的应用是支持 HTTP over SSL，即以 HTTPs 方式浏览 Web 安全网页(工作流程如图 13.16 所示)，例如网站的认证页面或电子商务、金融支付型网站的网页。此外，SSL 还可应用于 IMAP 安全邮件、构建 SSl-VPN 等业务。

在 HTTPs 安全访问中，采用不同于 HTTP 默认的 80 端口，而是采用专用端口号 443。

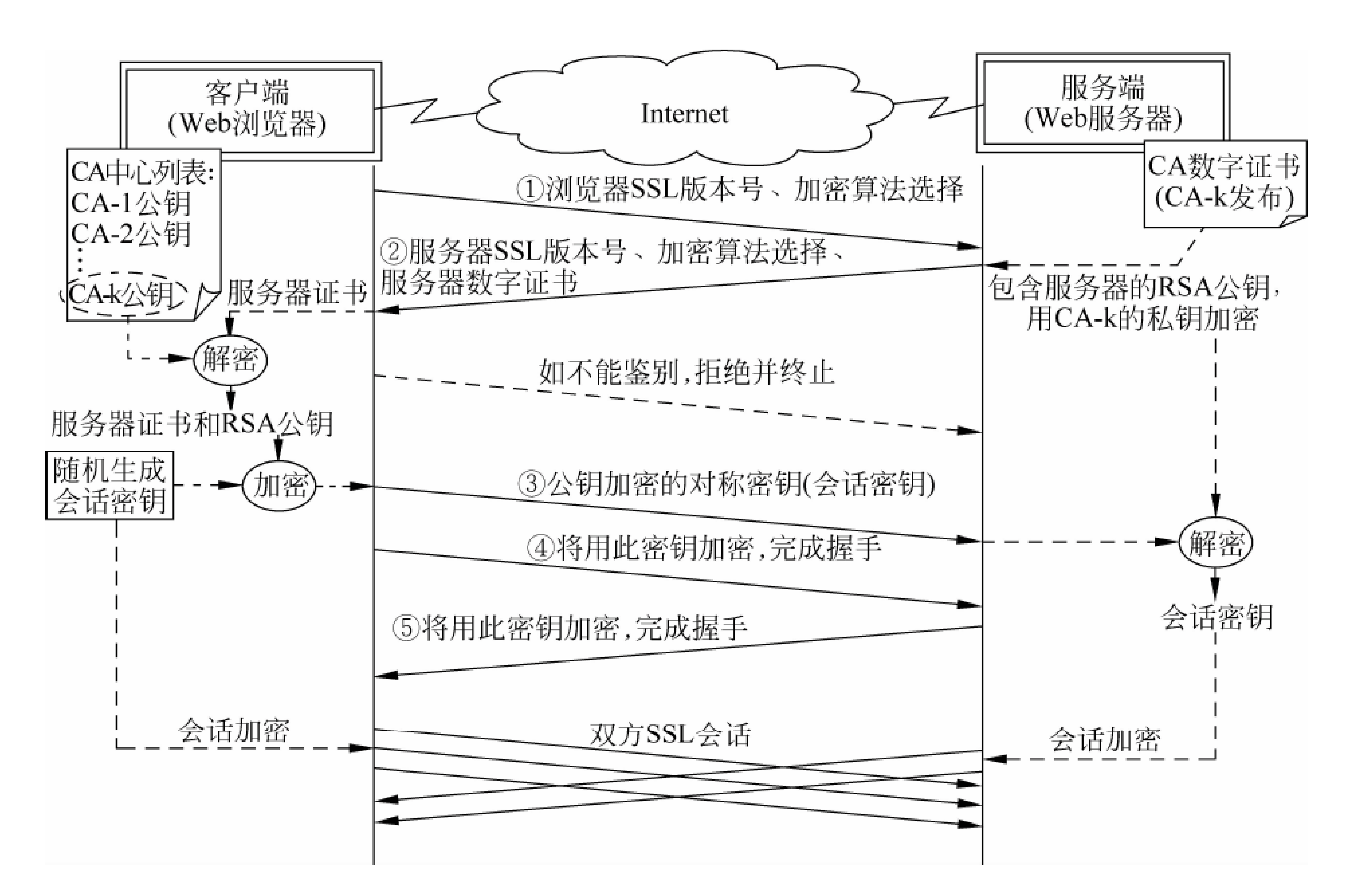


图 13.16 使用 SSL 浏览安全网页工作流程

13.4 WLAN 安全

13.4.1 WEP 协议

有线等效保密(Wired Equivalent Privacy,WEP)协议用于 WLAN 无线终端接入认证和数据加密,以防止非法用户窃听或侵入无线网络。

WEP 是 IEEE 802.11 标准的一部分,使用密钥长度可变的 RC4 流式对称加密技术实现保密性,并使用 CRC-32 校验技术保障信息的完整性。

WEP 采用两种安全认证方式:开放系统认证(open system authentication)和共享密钥认证(shared key authentication)。签证使用开放型的认证方式,只要输入密码,经验证正确即可获得通过。后者采用以下四个步骤进行验证。

(1) 接入终端向接入点(AP)发送认证请求。

(2) AP 回复一个明文消息。

(3) 接入终端用密钥(设置的字符串或十六进制数值串)对明文加密,再次向接入点发送认证请求。

(4) 接入点采用相同密钥对加密数据进行解密,比较明文消息是否一致,并决定是否接受请求。

由于 WEP 在密钥和认证方面存在安全性上的不足,容易受到攻击,2003 年被 WPA 协议所取代。

13.4.2 WPA 协议

无线局域网保护接入(Wi-Fi Protected Access,WPA)协议的作用与 WEP 类似,是对 WEP 安全技术的改进。2004 年形成 IEEE 802.11i 标准(又称为 WPA2)。

WPA 的数据加密采用**临时密钥完整性协议**(Temporary Key Integrity Protocol,TKIP)或 AES(Advanced Encryption Standard)对称加密算法,认证有两种模式可供选择:一种是采用 IEEE 802.1x 认证框架和**可扩展认证协议**(Extensible Authentication Protocol,EAP)的企业安全模式,另一种是称为**预共享密钥**(Pre-Shared Key,PSK)方式的个人安全模式,用于不需要设置认证服务器的家用或小型办公网络。

TKIP 仍然采用 WEP 所用的 RC4 加密算法,但加强了密钥的安全强度。TKIP 的密钥更长,达 128b,而且是动态变化的,即每个数据报文使用不同的密钥。其基本原理是:通过认证服务器或手工输入一个临时密钥用于会话,将临时密钥与每个站点的 MAC 地址进行混合,再加上 TKIP 序列计数器值、48b 初始化向量,产生 RC4 所用的加密密钥。

IEEE 802.1x 认证则更为严格,需要配置专门的认证服务器,运行 RADIUS 或 Kerberos 协议,提供集中式安全认证和接入控制。

13.4.3 WAPI 协议

无线局域网鉴别和保密基础设施(Wireless-LAN Authentication and Privacy Infrastructure,WAPI)是 2006 年由中国制定的无线局域网安全强制性标准(GB 15629.1101—1104),现已

获得国际标准化机构的认可，与 WEP、WPA 等并列为 WLAN 的安全保护技术。

基于 WAPI 协议的 WLAN 安全网络由无线终端、AP 和认证服务器（AS）三个实体组成，运用公开密码体系完成终端和 AP 间的双向认证，认证过程中采用椭圆曲线加密算法（ECC，192b 密钥），并协商生成会话密钥；通信过程中的数据加密采用国家密码主管部门指定的对称密钥加密算法（如 128b 密钥的 SMS4）。WAPI 支持在通信一定时间间隔或传输一定数量的数据包后，更新会话密钥。

WAPI 主要包括以下两个方面。

（1）无线局域网认证基础设施（WAI）不仅具有更加安全的鉴别机制、更加灵活的密钥管理技术，而且实现了整个基础网络的集中用户管理，从而能够满足更多用户和更复杂的安全性要求。

（2）无线局域网保密基础设施（WPI）对 MAC 子层的 MPDU 进行加解密处理，分别用于 WLAN 设备的数字证书、密钥协商和传输数据的加解密，从而实现设备的身份鉴别、链路验证、访问控制和用户信息在无线传输状态下的加密保护。

CHAPTER 14

第 14 章 网络安全威胁

14.1 网络安全威胁原理

随着 Internet 应用越来越普及、网络资源越来越丰富，网络安全问题也日益突出。然而，我们应该清醒地认识到，网络安全威胁并非刚刚才有，而是长期以来一直存在的。从来就没有绝对安全的网络，今后也不会有。因为只要信息在网络上流动，就有可能被窃取、篡改和伪造；只要计算机连接在网络上，就有可能被攻击、盗用、窥视和毁坏；只要数据和内容是有价值的，就有可能被人觊觎，不知不觉中遭遇黑手；只要系统有一丝漏洞，就有可能被入侵者利用；只要网络用户心存一丝侥幸或偶尔疏忽，就有可能酿成大灾难。

指望网络越来越安全是不现实的。或许现实世界也是如此。既然人无法生活在真空中，那就只有在弥漫着病菌的空气中想办法健康生活，比如增强免疫力、接种疫苗等。网络安全防范技术正是为网络与信息系统构筑的安全屏障。

但是，倘若为了安全而安全，就偏离了网络原本的目的。网络是为人类的沟通、交流、共享、合作服务的，是工作、学习、生活的工具。所以，不能为了安全而拖累网络的通畅、牺牲网络的效率、丧失网络的便捷，更不能为了所谓的安全对网络应用敬而远之。

面对危机四伏的 Internet，应该辩证地看待网络与信息安全问题。

首先应当承认并直面网络安全威胁的客观存在，而不是去否认或回避。其次应当使用有效的手段来防范安全威胁，保护网络与信息系统安全。

安全威胁与防范是一对共依共存的矛盾体。矛尖还是盾坚、魔强还是道高是永恒的话题。系统的安全性总是相对而言的：在时间、范围和程度上。网络系统在明处，攻击者在暗处，明枪易躲，暗箭难防，因此，保持持续的警惕性、具备充分的危机意识是十分必要的。

网络安全无小事。一些不起眼的小习惯、小失误、小细节，对信息安全而言都可能是引发大事故的导火索。网络系统是为人类服务的，两者是一个整

体。如果使用者随意、马虎,那么纵然有网络安全的铜墙铁壁也将是形同虚设。

网络安全防范是一项长期、艰巨的任务,需要良好的技术与不懈的努力。但安全防范系统的投入通常较高,运行维护工作量也较大,因此,应该全面而深入地研究网络安全技术,知己知彼,才能有的放矢地部署安全技术,寻找攻守平衡点,把握系统关键之所在,使技术发挥最大的效益。

如图 14.1 所示,网络安全威胁来自计算机网络体系的各个方面、各个层次、各个环节,是立体的、全方位的、无处不在的。只有充分了解网络安全威胁的类型、原理、方法、工具、来源、条件、特点等具体情况,才能进一步实施有效的防御工作。但网络安全威胁技术繁多,而且一直在变化发展中,无法完全遍历,但是,可以从已知的网络安全威胁技术中研究其本质的、共性的技术,从而对已有的安全威胁有比较全面的把握,对未来未知的安全威胁也能起到举一反三的认识作用。

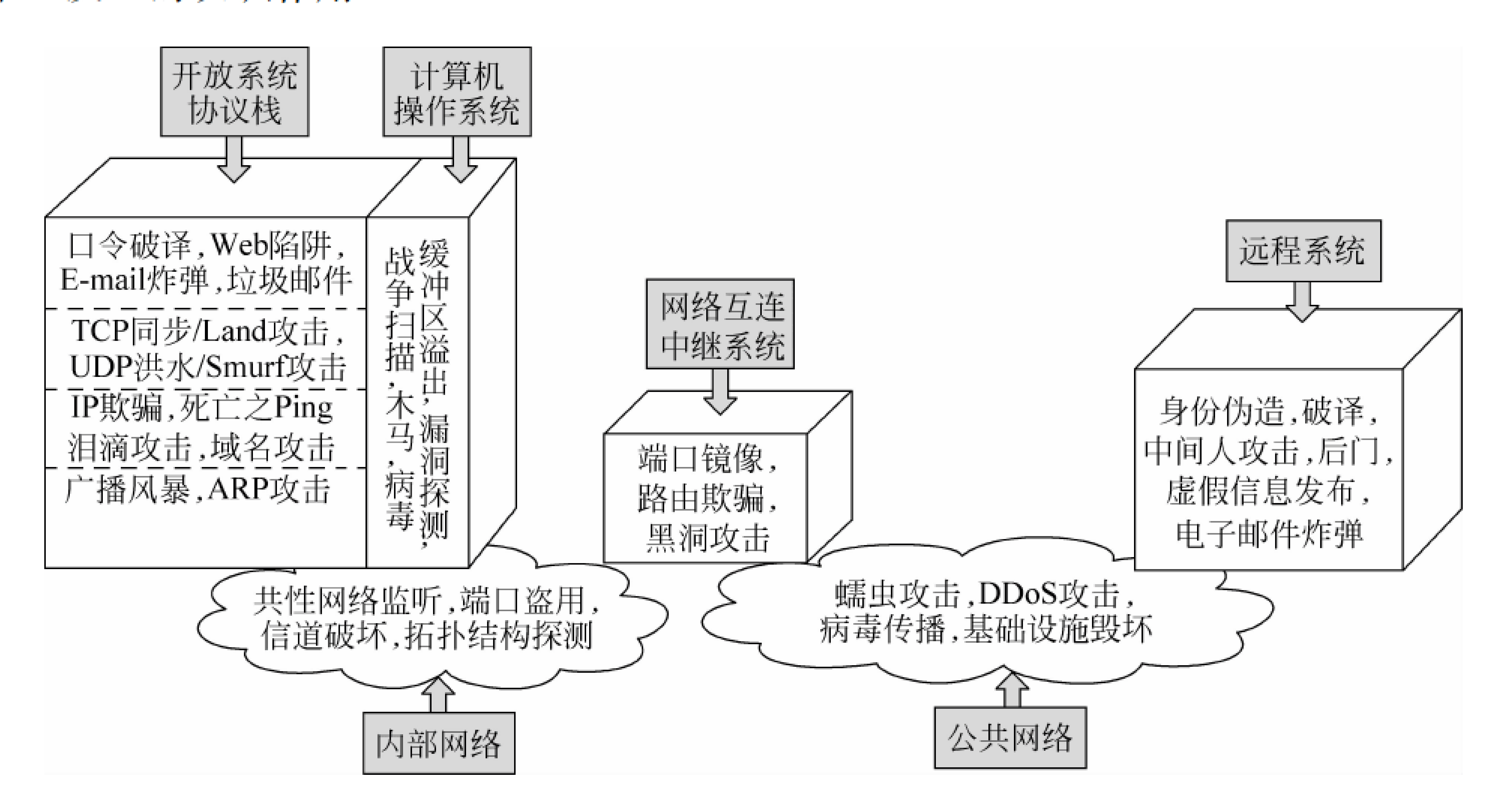


图 14.1　网络安全威胁方位

我们常把网络安全攻击者称为黑客(hacker),但黑客原意指醉心电脑技术的高手,并非贬义词,对恶意入侵者实际上有一个专门的名词叫骇客(cracker)。

14.2　网络攻击基本技术

14.2.1　通信监听

窃听的历史大概与人类同龄。在计算机网络领域,获取通信数据并对内容和协议加以分析,可以直接或间接地掌握各种信息,因此通信监听是网络安全攻击的一项基本技术。

一些专用的协议分析设备,不但能够抓取并记录完整的通信过程,而且可以根据协议还原出信息内容。例如,明文传输的用户名和口令;用 FTP 下载的文件;用 SMTP 传输的电子邮件等。

通信监听往往是各种安全攻击的第一步。依据网络结构、技术、环境、条件等不同情况以及攻击目的的差异,通信监听(也称旁路监听)有多种可行的技术方法。

(1) 线路插入法：通过T型接插件搭线等手段，把正常通信信道上的信号引出，使监听设备可以接收到通信双方的所有数据。监听者并不发送数据，所以通信双方无法感知其存在，具有很强的隐蔽性。与从电线上偷引电力、从管路盗窃油气等不同，信道上的数据并不会因为监听而受损。

(2) 共享网络法：利用共享信道网络的特性，如WLAN、Ethernet Hub等，只要接入共享域就可以侦听到所有设备的通信数据。尤其对于无线通信，监听活动可以做到神不知鬼不觉。

(3) 端口镜像法：支持端口镜像(port mirror)功能的网络交换机、路由器可将一个或多个端口的通信数据复制到监听端口(可为虚拟端口)。如果通过远程登录等方式控制网络设备启动端口镜像，即可实现对指定设备(端口)的监听，且可以在远程悄然接收转发过来的数据。

(4) 嵌入程序法：在被监听对象系统中植入监听程序，可采集用户键盘输入、屏幕显示内容及通信数据，利用网络传递给监听者。潜伏的程序非常隐蔽，不像线路插入等物理方法那样容易被察觉。但是，如何嵌入监听程序在技术上有相当难度。

在网络软件系统调试、测试中，程序员往往会添加一些特定的代码，用于监听会话过程，便于定位错误。如果这类代码无意或有意忘记清除，则可能成为用于窃听的后门。

(5) 电磁泄露法：计算机与网络运行过程中会发出各种电磁波(电磁泄露)，采用特殊的监听设备可以接收并还原数据。

14.2.2 漏洞扫描

安全攻击者利用专门的工具对网络与信息系统进行尝试性的探测，寻找可攻击的弱点、隐患、缺陷，称为漏洞扫描(vulnerability scanning)。

漏洞扫描是网络安全攻击的前奏曲之一。攻击者面对看似密不透风的网络体系，首先要探寻可能的入手点，然后才能进行针对性的攻击。

不幸的是，绝大部分的安全漏洞是已知的。例如，操作系统已公布的缺陷、协议机制上的缺陷等。但是，因为专业知识、重视程度、响应周期、技术能力上的各种原因，用户并不了解自身安全漏洞、没有及时弥补安全隐患，而攻击者或扫描工具则对各种安全弱点掌握得很全面，从而存在信息不对称和操作时间差。如果漏洞恰好被捕捉到，那么被攻击的厄运就在所难免了。

漏洞扫描的基本原理是试探询问、分析响应。例如，不同的操作系统(或操作系统不同版本)在实现某些协议时稍有差异，对一些异常报文，有的处理方式是予以丢弃，有的则回复拒绝报文，有的发送断链报文。因此，通过发送特别设计的报文，根据系统的不同反应，可以判别系统类型或其版本，进而利用已知的安全漏洞。

漏洞扫描的主要方式有以下几种。

(1) 扫描网络互连设备，特别是安装在内部网络计算机上的电话网调制解调器，可以直达内网纵深处。这一探测所有可能存在的接入点的方法被形象地称为战争拨号。

(2) 端口扫描(port scanning)用于了解网络体系开放的TCP/UDP端口，以便了解其安装的应用种类，或使用某些端口号实施攻击。

(3) 实施网络体系结构的探测可掌握内部网络的规模、IP地址分配情况、设备分布(尤

其是服务器分布)、子网划分、安全措施等,确定攻击范围和目标。

(4) 操作系统漏洞检测可使攻击者利用用户尚未对操作系统公布的漏洞打补丁的真空期实施精确攻击。

14.2.3　口令破解

口令是绝大部分信息系统的第一道安全防护措施,事实上,甚至是许多系统的唯一认证手段,例如操作系统、电子邮件、无线基站、信用卡等。简单的口令背后隐藏的是资源、隐私或财富,难怪网络入侵者对破解口令趋之若鹜,因为只需突破这一项保护,系统就会门户大开。

口令破解是安全攻击最基本、最常用的技术。

(1) 监听法:使用通信监听工具侦听合法用户与服务器的通信过程,尤其是登录过程,通过分析获取用户名、口令等重要信息。

(2) 重放法:对于非一次性口令登录机制,攻击者不需要破译获取的加密口令,只需将记录下来的加密口令直接发送,即重放(playback),就可实现登录。

(3) 陷阱法:攻击者模拟合法服务端(如网站),骗取用户登录,以获取用户账号和口令等重要信息。陷阱法适合非指定目标用户的攻击,可能批量获取大量账户信息。这一方法与其他手段结合后攻击性更强,如:篡改网页合法 URL、通过病毒修改用户计算机 hostid 文件(静态域名解析列表)等。陷阱不限于在网络上,例如通过开设店铺盗取刷卡客户的信用卡账号信息。

(4) 猜测法:许多用户从不修改千篇一律的初始口令,也有大量用户采用不安全的口令,容易给口令猜测者造成可乘之机。空口令、与用户名相同的口令、1111、123456、asdf、日期等都是口令猜测者首先会去尝试的,而且具有相当高的命中率。

思考:*6 位或 8 位数字的口令,假如设为生日,试计算猜中概率。*

(5) 字典法。口令字典(又称黑客字典)是一个用各种常用单词、词组组成的列表,数量大大地小于随机组合的 n^{26} 或 n^{36}(设口令长度为 n),而且便于软件实现自动的依次尝试。字典法的原理是基于揣测用户心理和习惯,因为绝大部分用户不会去使用毫无意义的字母、数字的组合。

思考:*服务器可以采用何种手段应对字典法猜测口令?*

(6) 逆向法:利用某些系统的特性,不采取直接破译口令的办法,而是用一个简单的口令尝试账号的组合。在用户数量较大的情况下,命中率很高,适合于非特定目标用户的攻击。例如,在大型电子邮件系统中,通过简单的用户名字母组合,总能找到一批用户的口令是 888888。

(7) 穷举法。穷举法(或称为暴力破解)是口令破解者迫不得已在技术上所用的最后一招,逐一尝试用户口令的所有组合,虽然是最为简单粗暴的办法,但只需假以时日,必然能实现破解。

思考:*登录系统可采取什么针对性措施对抗穷举破解法?*

(8) 间接法:用户账户的盗取者采用迂回策略,以达到与获取口令效果相同的目的。例如,通过普通账号获取更高权限账号(如溢出攻击);通过了解某些编程人员有意或无意设置的软件后门(如万能口令)来突破;通过破解系统中安全防范薄弱的计算机再转移到真

正的攻击目标。

(9) 物理法。物理法是指采用现实世界的犯罪手法获取口令,例如,潜入办公室用内部网络终端窃取信息(内部防范往往较为松懈)、盗走保存账户信息的磁盘、贿赂管理员以获得用户信息、绑架用户逼迫其说出口令、从黑市购买账号等。

思考: 如何设计灵活、安全、易记的口令?

14.3 恶意代码攻击

14.3.1 病毒

自然界的病毒、细菌、寄生虫对人类等各种生物的健康造成很大危害,有些危害具有长期的潜伏性,非常隐蔽,有些危害甚至直接造成生命的终结。计算机和网络中类似病毒感染宿主并进行攻击的现象同样存在,同样危害巨大。Internet 的普及使感染攻击的波及范围更大、传播速度更快、变化种类更多。

病毒(virus)是一种恶意代码(malware),是计算机网络的毒瘤。病毒的历史可以追溯到计算机单机时代。

病毒是一种特殊的、精巧的计算机程序,具有如下特点。

(1) 寄生性。病毒程序并不单独存在,而是附着在其他软件上。通过修改宿主程序原有的运行流程,强行附加病毒代码,达到运行自身的目的。

(2) 复制性。病毒程序的重要目的和能力是复制自身,就像现实世界的生物病毒一样,千方百计地让自身副本越来越多。

(3) 感染性。病毒不断检查指定类别的文件,特别是可执行文件(.com、.exe 等),一旦发现对象是干净的,立即实施感染。

(4) 破坏性。病毒都会对计算机和网络系统造成不同程度的损害。

病毒的危害性在于浪费存储空间、干扰正常运行、消耗计算能力、降低程序效率,严重时将引起数据丢失、文件毁坏、服务中止和系统崩溃。为了防范病毒的投入更是一笔巨大的开销,也因此增加了系统的复杂性和维护难度。

从病毒存在的特性和状态来区分,有以下类别。

(1) 在野病毒(活跃病毒):仍然在传播并感染用户的病毒。如果被记载但不再传播的病毒,相应称为非活跃病毒(not in the wild)。

(2) 实验室病毒(动物园病毒):病毒存在于可控环境,不会传播到外界。

(3) 同伴病毒:病毒对目标程序进行重命名,而自己使用目标程序的名称,当目标程序被执行时,病毒得到运行。

(4) 空腔(cavity)病毒:病毒把自己隐藏在充满大量数据 0 的宿主程序中,不增加原程序大小、不破坏原程序代码,难以被发现。

(5) 隧道(tunnel)病毒:病毒通过在操作系统中寻找中断处理程序(免检内核区),将自身安装在防病毒软件后面,逃避监测。

(6) 直接行动病毒(非常驻病毒):每次宿主程序被执行时,病毒只感染一个或多个程序,而非尽可能多的文件,使用户不易察觉异常。

（7）内存(RAM)病毒：当被感染程序运行时，病毒将自身单独转移到内存中(不会随宿主程序停止运行)，可持续感染其他被执行程序。

（8）隐秘型病毒：病毒使用各种手段隐藏自身踪迹。

（9）潜伏病毒：病毒隐藏在系统中，直到满足某些条件才激活并爆发。

（10）多态病毒：病毒经常改变外形，每次复制时都改变配置。每个副本都是原来病毒的一个变体。工作原理相似，但代码标记各不相同，难以通过特征代码来检测。病毒还使用各种加密、压缩方法来逃避检查。

如果按病毒感染的不同对象类型来区分，有引导扇区病毒、文件感染病毒、DOS 病毒、Windows 病毒、宏(macro)病毒、脚本(script)病毒、Java 病毒(病毒因此具备了跨操作系统运行能力)、Shockwave 病毒(flash 病毒)、复合型病毒(混合型病毒)等。

14.3.2　木马

古希腊特洛伊木马(trojan horse)的传说十分脍炙人口，因象征着智慧、狡黠、技巧、勇气、想象力、出其不意与克敌制胜而载入史册、千古流传。而 Internet 上的特洛伊木马则让这个传奇蒙羞，是邪恶、阴险、奸诈的代名词。

特洛伊木马(简称**木马**)技术原理及所起的作用与古老的木马非常相似，只是化身为一种间谍程序(spyware)，属于恶意代码类型之一。木马的特点是：

（1）利用某种载体(如网页、电子邮件)侵入用户计算机系统；

（2）隐蔽性地驻留与执行，平时难以察觉；

（3）通过某种设定条件触发激活，完成指定的攻击任务。

木马通常作为攻击过程的一个环节，起到里应外合的作用，因此木马也属于植入式后门(backdoor)程序。木马的功能可分为监听、控制、跳板、破坏等。

木马分为客户端和服务端两个部分，其中客户端是攻击者用来远程控制木马的系统，服务端(又称为守护进程)即是木马程序。被植入木马的计算机即成为傀儡机、僵尸机(俗称肉机)。

当木马服务端程序成功潜入计算机，并通过特定的启动方式运行后，木马程序就会打开某个 TCP/UDP 端口，在该端口上侦听是否有连接请求，随时等待客户端(攻击程序)的远程控制命令。对客户端而言，如果网络上有大量计算机被“种马”，就可以随时发号施令，可在同一时间调动千军万马，对网络实施分布式攻击(例如拒绝服务攻击)。

木马隐蔽性很强，比较难以发现；非常具有迷惑性，发现后难以定位；还具有相当强的顽固性，即使被定位也很难彻底清除。斩草却不能除根，往往让人望马兴叹。原因是木马采用了多种对抗清除的技巧。

（1）木马使用多种启动方式：随操作系统启动、随文件运行启动、随动态链接库调用启动等。

（2）木马经常刻意变换文件名，或采用与操作系统核心程序非常类似的文件名，借以伪装自身。同时修改文件显示和访问属性、伪装图标，混迹于系统内核程序中。

（3）木马驻留在操作系统的各个位置，如系统文件、应用程序、注册表、启动项、服务组等。

（4）木马在内存或进程表中也隐蔽得很深，一般权限的用户使用常规的命令甚至无法

发现该进程的存在。

木马需要通过一定的方式进入系统。实际上,在 Internet 中这样的机会是很多的。常见的有随 E-mail 及其附件传播、附加在盗版软件中、诱骗用户下载软件执行等。与病毒不同,木马是以独立文件形式存在的。为麻痹用户,木马总是伪装成 HTML、TXT、ZIP、JPG 等文件(修改扩展名或图标),当用户看到这类文件时,很容易误认为一般的文档而打开(运行)。木马也会利用某些操作系统存在的漏洞,通过 Script、ActiveX 及 ASP、CGI 交互脚本的方式植入。

木马的激活方式主要有两种:①攻击程序发送命令方式;②满足预先设定的某个条件(如日期或时间、窃取到账号和口令)。

一个功能强大的木马一旦被植入系统,攻击者就可以像操作自己的机器一样远程控制系统,自动筛选、监控所有操作。例如"冰河木马"(端口号 7626),可以实现自动跟踪目标机屏幕变化、完全模拟键盘及鼠标输入、记录各种口令、获取系统信息、限制系统功能、远程文件操作、注册表修改等。

14.3.3 蠕虫

蠕虫(worm)是一种在 Internet 上传播的攻击程序,是恶意代码的形式之一。蠕虫的始作俑者是 Robert Morris。1988 年,他发布了一段专门攻击 UNIX 系统缺陷、名为 worm 的代码,故名。该程序仅区区 99 行,却拥有巨大能量。程序用 finger 命令搜索联系人名单,然后用 E-mail 向名单上的用户复制、传播蠕虫本身。据统计,在短短数小时内,使当时 Internet 上的 6 万~8 万台计算机的 10%~20%陷于瘫痪,损失惨重。

最初有一些技术人员探讨网络蠕虫的设计,设想由一个特殊程序在计算机间游荡,当发现有计算机负荷过重时,该程序可以从空闲计算机借取资源而达到系统的负载平衡。莫里斯蠕虫不是借取资源,而是耗尽资源。

蠕虫有别于病毒,但可看做一种特殊的病毒。两者都是通过复制自身来达到传播和破坏的目的,但普通的病毒需要宿主来寄生,而蠕虫则是在网络上独立存在和运行的。

一个被蠕虫感染的计算机系统表现出的行为与受到木马感染的系统很类似,如出现不寻常读写磁盘动作、运行缓慢、出现多个未知任务、配置文件和注册表被修改、E-mail 被自动发送等。

蠕虫的传播方式也与木马采用的相同:通过 E-mail 附件、即时通信客户端或计算机用户之间共享文件、安装不明来源文件等进入系统。然而,蠕虫可以不在用户参与(用户不知情)下自己传播。例如,早期版本的 E-mail 客户端软件具有自动预览功能,用于显示附件内容方便用户阅读,这使得蠕虫可以在用户不主动干预执行的情况下其代码也能被激活,并立即向地址簿中所有用户发送包含蠕虫的邮件。

由于蠕虫具有自发获取地址进行发送的功能,传播力非常惊人,可以呈指数型进行扩展,严重消耗网络带宽和计算资源。但是,蠕虫也有自身的弱点,一旦切断蠕虫的网络传播途径或自动运行机制,蠕虫就会销声匿迹,而不会像病毒或木马一样驻留下来。因此,网络用户应该注意甄别收到的信息,不要点击打开可疑邮件或消息,就能有效阻止蠕虫继续蔓延。

CHAPTER 15

第15章 网络安全攻击

网络与信息系统是由各种软件、硬件设备构成的复杂的综合体。软件包括操作系统、数据库系统、应用系统，其中包含各层次的协议机、各类服务接口等。硬件则包括计算机、网络设备、通信媒介（信道）等资源。构成系统的每一个组件、每一个接口、每一个环节，都可能由于设计上或实现上存在的漏洞，被非法利用，引发相应的网络安全攻击。

我们不得不承认，许多网络安全攻击技术具有极高的技术含量，算法十分巧妙，散发着智慧的光芒。可惜聪明没有用在正道上。我们学习这些技术，当然不是为了效仿恶行去攻击别人，而是可以加深对网络和信息系统的认识，并且了解自身不足，有针对性地加以弥补和完善。

15.1 缺陷攻击

15.1.1 拒绝服务攻击

拒绝服务(Denial of Service，DoS)攻击是一种很常见，也很特殊的网络安全攻击类型。实施拒绝服务攻击时，攻击者不必事先获得对系统的控制权，也不是为了获取系统访问权，其目的是使系统或网络过载，使之无法继续提供正常服务。

DoS攻击可分为两种不同类型：带宽损耗型（bandwidth consumption）和资源匮乏型(resource starvation)。前者因大量占用系统（例如一个网站）的网络带宽，导致受害者系统无法向其客户提供正常服务；后者则使目标服务器因宕机或没有足够资源（如内存）而失去网络服务能力（无法正常建立连接、响应请求、传输数据等）。

DoS攻击是攻击者系统与受害者系统拼资源、拼性能的过程。攻击者如果没有足够带宽和计算能力，就无法耗尽攻击对象的系统资源。所以有效的DoS攻击一般采用**分布式拒绝服务**(Distributed DoS，DDoS)方法，即在同一时间调用大量不同位置的计算机向同一目标实施攻击（如图15.1所示）。为

此,DDoS 攻击需要召集大量的傀儡机(puppet node),协助攻击行为。傀儡机是被事先植入了恶意程序(如木马)的计算机,平时处于潜伏状态,一旦需要时即可通过一定方式激活,受控参与指定行动。

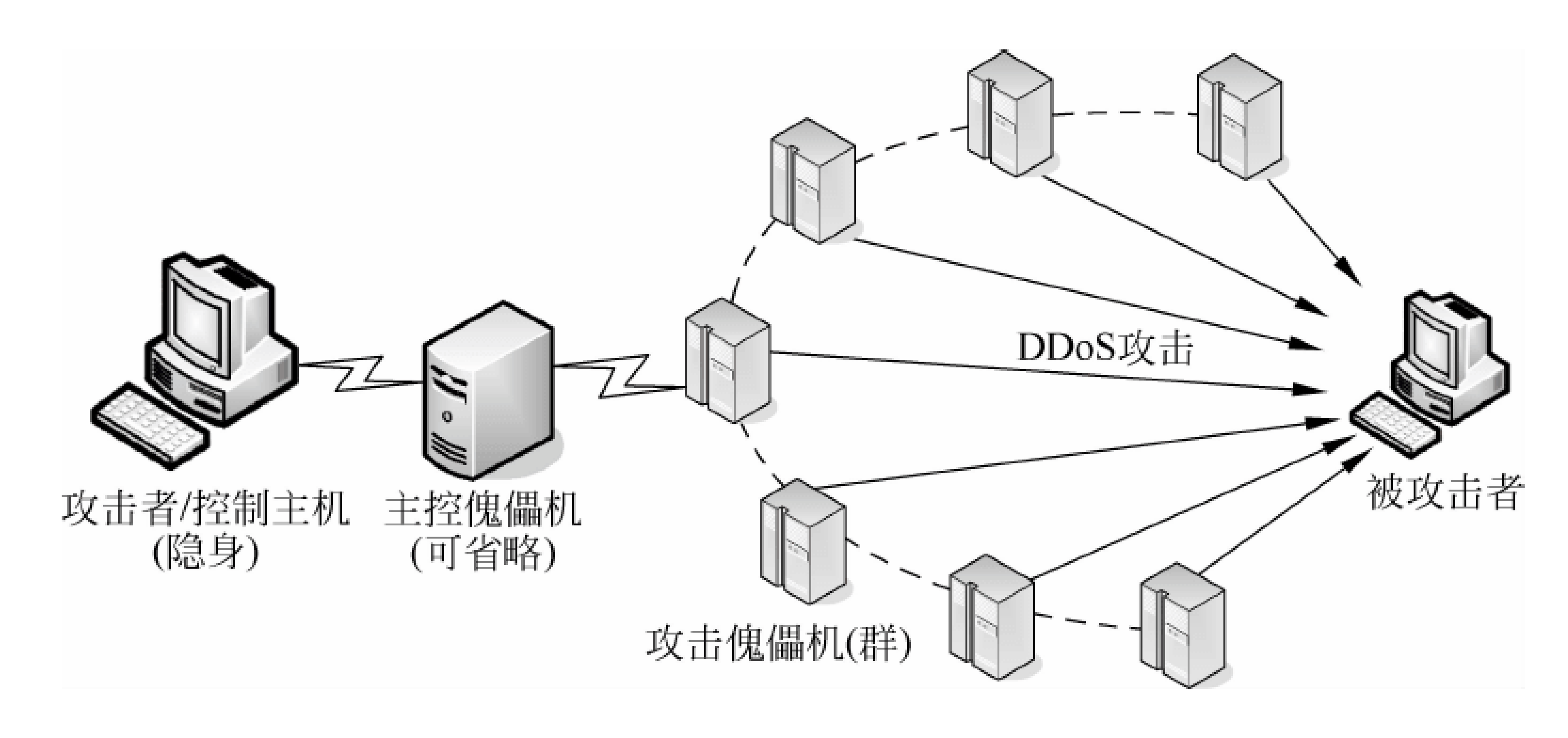


图 15.1　DDoS 攻击原理

DDoS 攻击方法很多,可分为三种技术类型。

(1) 利用协议漏洞:Syn Flood、Smurf、Fake IP。

(2) 利用软件缺陷:Tear Drop、Ping of Death、Land、ICMP Fragment。

(3) 进行资源比拼:ICMP Flood、UDP Flood、E-mail Bomb。

1. Syn Flood 攻击

同步洪水(Syn Flood)攻击也称为 TCP 同步攻击、三次握手攻击、半连接攻击。同步攻击是最典型的 DoS 攻击手段。

同步攻击基于 TCP 建立连接所用的三次握手机制(如图 15.2 所示)。攻击者的思路就是故意缺少第三步的 ACK 回复(图中的阴影部分),使被叫方计算机的连接一直处于占用资源的等待状态,直到计时器超时释放。如果同时存在大量的半连接,计算机就会耗尽系统资源,无法为正常业务提供服务。

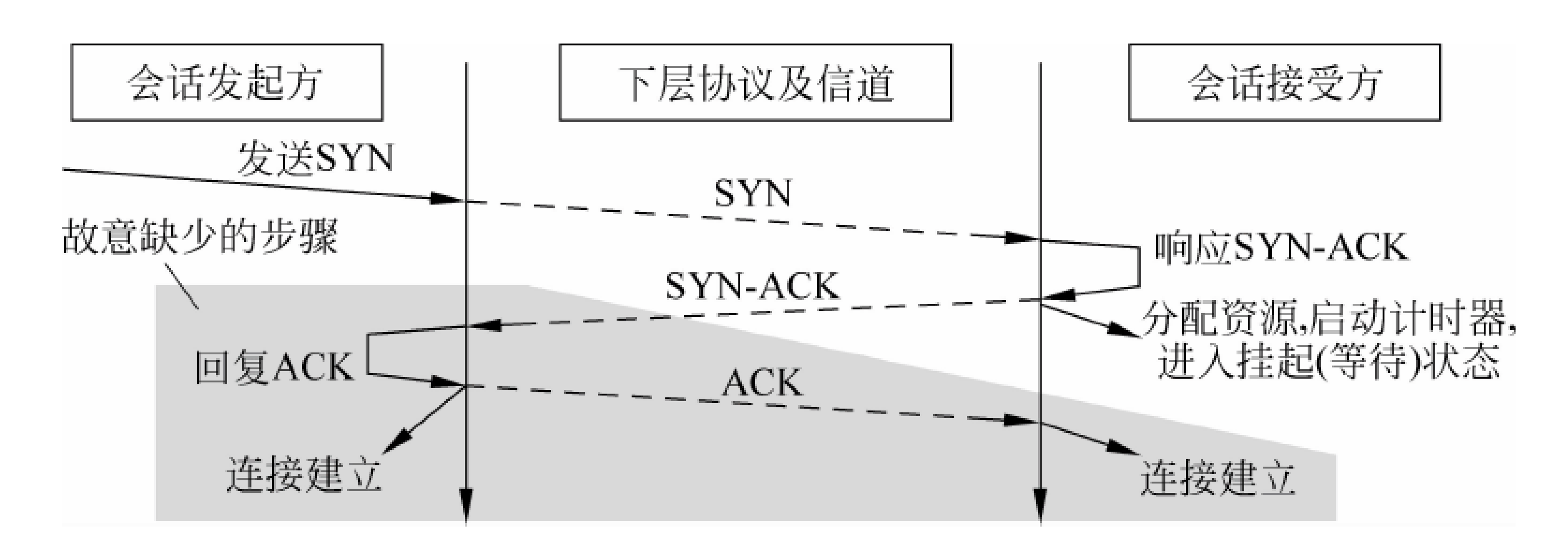


图 15.2　同步攻击原理

同步攻击发起者会使用不同的 IP 地址,使接收方难以甄别不正常的流量;大量的 SYN 几乎同时从傀儡机群发出,以便在连接等待计时器超时前突破系统资源上限;同步攻击还可变化为 **Land 攻击**,即把 SYN 报文的源和目的 IP 地址均设置为被攻击者的 IP 地址,使之向自己回复 SYN-ACK 和 ACK 报文,可能由此建立起自环的空连接,占用更长时间和更多

资源。

2. Ping of Death 攻击

死亡回显(Ping of Death)利用 ICMP 服务端协议机实现的缺陷实施攻击，是一种 DoS 攻击的方法。例如某操作系统规定 ICMP 报文最大尺寸不超过 64KB，却不进行严格检查，攻击者故意发送超过 64KB 上限的畸形 ping 报文(基于 ICMP 回显功能)，主机就会出现性质严重的内存分配错误，导致崩溃、死机。

向目标主机长时间、连续性、大批量地发送 ICMP 报文，也可能致使系统瘫痪。大量的 ICMP 报文形成 ICMP 风暴，使得主机耗费大量的计算资源，疲于奔命。

如图 15.3 所示，死亡回显的一种变化是 **Smurf 攻击**：向一个拥有大量主机的内部子网发送欺骗性的 ping 报文，目的 IP 地址设为该子网的广播地址(主机号全 1)，而源 IP 地址设为子网内被攻击主机的地址。早期版本的某些型号路由器在接收到该 ICMP 报文后会予以转发，结果是向子网内发送了广播报文；然后所有主机都会收到报文，根据协议要求都向指定的目标回复报文(即 echo)；目标主机将同时收到大量 ICMP 报文。如果子网规模较大，很可能导致目标主机失去正常服务能力，达到 DoS 攻击的目的。如果反复使用 Smurf 攻击，还可能使子网因广播风暴而瘫痪。

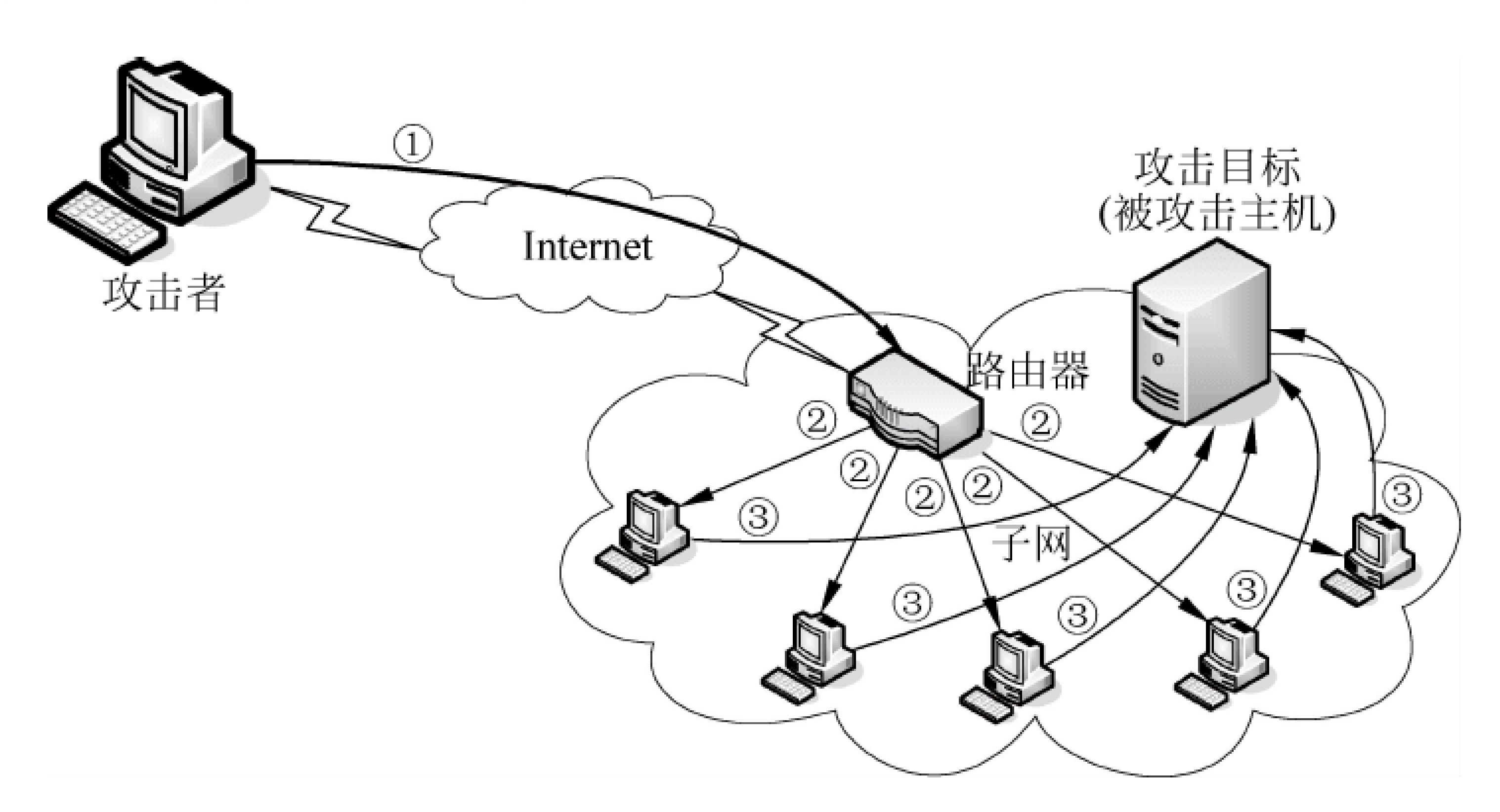


图 15.3　Smurf 攻击原理

Fraggle 攻击与 Smurf 类似，区别是使用 UDP 的 echo 功能。

死亡回显的另一种变化是 **UDP Flood 攻击**。攻击者随机地向被攻击系统的端口发送 UDP 数据报文，就可能形成 UDP 洪水淹没式攻击。当目标系统接收到一个 UDP 数据报文，会根据目的端口号试图确定正在等待(侦听)中的应用程序，如果发现应用程序并不存在，就根据报文中源 IP 地址回复一个 ICMP 报文，报告"目的地址无法连接"。那么，如果向目标主机的随机端口不断发送随机端口号的 UDP 报文，在忙于处理这些垃圾数据报文的过程中，主机性能不断下降，直到不能提供正常服务。

基于 UDP 的攻击方式还可进行变化，攻击者利用服务器上开放的 UDP 的 chargen 和 echo 服务来发动 UDP Flood 攻击。chargen 服务用于测试目的(即 character generator)，端口号为 19，可对任何接收到的 UDP 报文反馈随机生成的字符；echo 服务端口号为 7，与 ICMP 相同，可对任何接收到的数据原样原路返回。于是，如图 15.4 所示，攻击者伪造 UDP 报文，发往某一开放 chargen 服务的主机，回复地址(源 IP 地址)指向开放 echo 服务的另一台主机，这

样两台主机之间就会无限制地往来发送 UDP 报文，生成大量的无效数据流，导致网络堵塞。

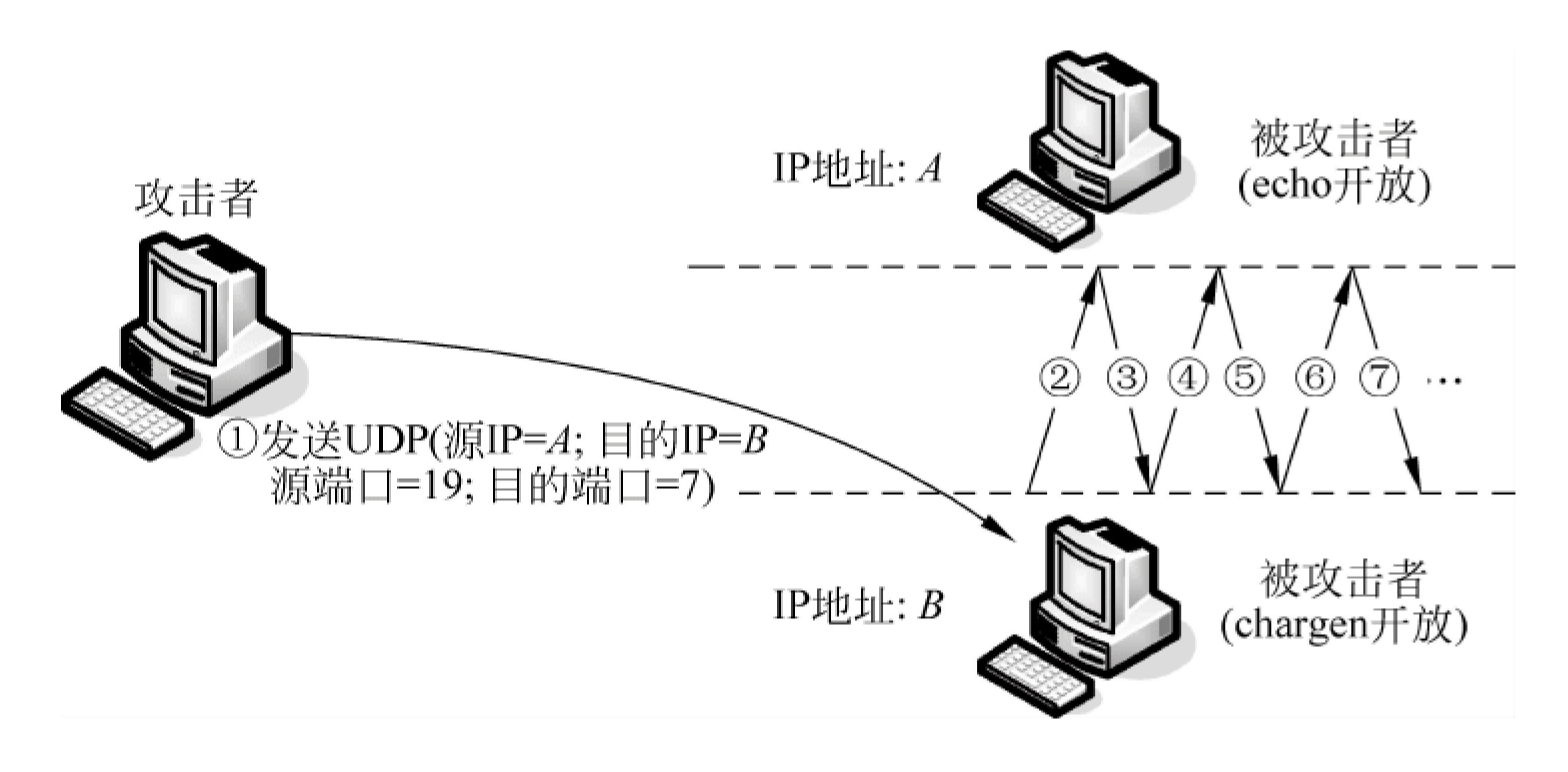


图 15.4　UDP Flood 攻击原理

3. Tear Drop 攻击

泪滴攻击(Tear Drop)是利用某些 TCP/IP 实现中分段重组的进程的潜在弱点实施的 DoS 攻击，属于利用协议机缺陷进行的畸形消息攻击技术。

在 TCP/IP 协议栈实现中，总是假定可以信任 IP 报文的控制头所包含的信息。某些 TCP/IP 协议机(存在于一些操作系统相关版本中)即如此认为。一旦收到含有重叠偏移量的分段数据，会因束手无策而崩溃。攻击者据此篡改或恶意设计 IP 报头，形成错误偏移量指针(offset 值错位)，发送给目标主机，引起主机宕机而终止服务。

很难追溯 Tear Drop 这个名称的由来。报文片段与泪滴实在难以在形象上联系到一起。难道是源于受害者因主机瘫痪、损失惨重而泪水涟涟？

4. E-mail Bomb 攻击

电子邮件炸弹(E-mail Bomb)攻击主要针对电子邮件服务器系统，即在同一时间内调用大量电子邮件服务器对同一个目标发送巨量电子邮件，使得该电子邮件服务器无法正常对外服务，或因超出其处理能力而崩溃，实现 DoS 攻击。

思考：如何调用电子邮件服务器？如何让电子邮件服务器向指定地址发送大量邮件？

5. Fake IP 攻击

IP 欺骗(Fake IP)攻击是指利用虚假 IP 报文(如伪造 IP 地址或其他字段、伪造 IP 承载的 TCP/UDP 报文)实施的 DoS 攻击。这种攻击技术也是其他攻击方法的组成部分之一。

IP 欺骗攻击利用 TCP 的 RST 位来实现。假定有一个合法用户(IP 地址为 A)已经同服务器建立了正常的连接。攻击者构造攻击的 TCP 报文，把 IP 源地址设定为 A，向服务器发送一个带有 RST 置位的 TCP 报文，使服务器误解而释放已有连接。这样，合法用户发送数据就因服务器失去连接而无法成功，不得不重新建立连接。为提升攻击效果，端口号应设置为服务器的重点应用，例如 Web 服务器的 80 端口。

攻击者可伪造大量的 IP 地址(例如该网段的所有主机地址)，向目标主机发送 TCP(RST 置位)报文，使服务器无法对合法用户进行正常服务。

15.1.2　缓存区溢出攻击

缓冲区溢出(Buffer Overflow)攻击不是让计算机的缓冲区出现问题那么简单,而是故意使核心代码的缓冲区发生溢出,进而接管操作系统的 root 控制权限。缓冲区溢出攻击的设计思路非常精巧,但也危害巨大。莫里斯的蠕虫即利用了这一原理,为如今网络上泛滥的恶意程序打开了一扇黑暗的大门。

缓冲区溢出错误是一种非常普遍、非常隐蔽,也非常危险的漏洞,在各种操作系统、应用软件中广泛存在。如今大约 80%的安全事件与溢出攻击有关。

思考:为什么程序代码不能完全(直接)反映缓冲区溢出错误?

从软件技术上说,如果对缓冲区的边界控制不严格,甚至不加以控制,那么一旦越界,就会使数据覆盖其他数据区域,引起不可预料的错误。例如 UNIX 系统中的 strcpy(字符串复制)函数,只检查 0x0 结束符,对字符串长度不作判别,如果字符串复制长度超过程序定义(申请)的空间,就会破坏其他区域的数据。那么,倘若对缓冲区溢出加以利用,就可以实现导致程序运行失败、系统宕机、重新启动等 DoS 攻击,或者使一个普通用户账号有机会升级为 root 账号,获得一台主机的全部的控制权。

除了 strcpy 外,实际上还存在别的危险函数,如 sprintf、gets、scanf 等;存在危险函数的操作系统当然也不限于 UNIX。

考察 UNIX 程序空间内存分配和进行函数调用时的栈数据结构(如图 15.5 所示)。设想某个函数的一条指令对图中本地变量 LocalVar *k* 作一次超长的数据复制操作,内容为一连串的字母 A(ASCII 编码 0x41)。该冒失的操作将依次破坏其他本地变量、被保护的 EBP、返回地址和函数实参寄存器、命令行参数等,其中最严重的后果是覆盖了函数的返回地址,使返回地址变为 0x41414141。这样,函数执行完毕后将"返回"到该错误的地址,而不是按正常逻辑返回程序的调用点,程序因此完全失控。类似的修改栈内返回地址的操作还可以通过精确的程序代码控制来完成。这就有了想象空间:假如把返回地址覆盖为预谋的地址,或许就可执行预谋的程序。

下面用一个缓冲区溢出攻击的实例说明缓冲区溢出攻击的原理。从实例中可以观察到系统的特性如何被利用、如何改变正常的程序流程、访问权限是如何改变的。实例的目标是

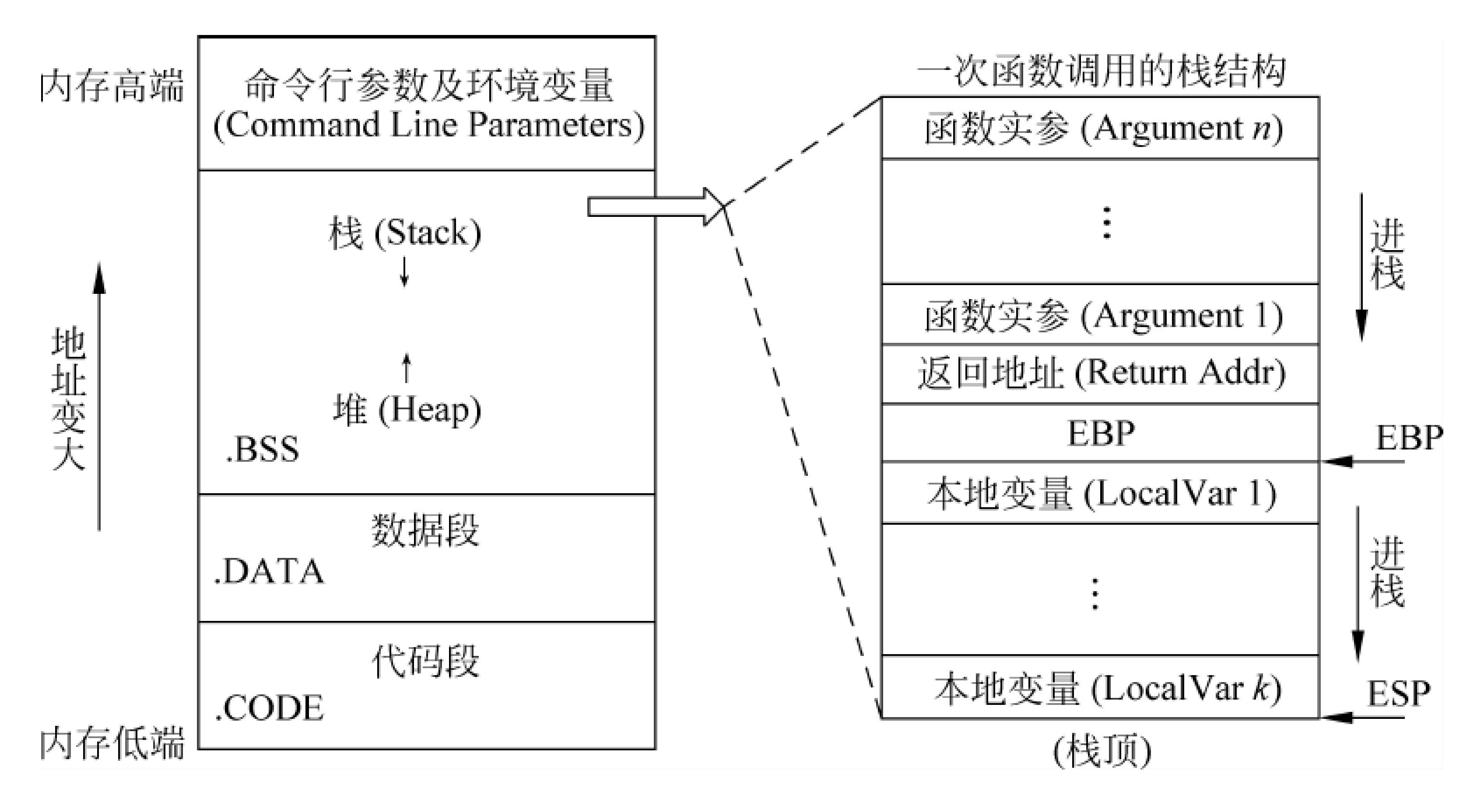


图 15.5　程序空间与函数调用栈结构

从一个较低的 UNIX 用户权限(如 user shell)获取最高的权限(root shell)。

在 UNIX 中,一个在低级别 shell 下执行的程序,执行完毕仍然回到原有级别,即使重复执行 shell 也不例外。但从微观的角度考察,程序执行过程中,调用某些系统程序(如系统库函数、软中断)时,将由系统权限(root 权限)接管控制,正常情况下,中断返回后自动恢复原 shell 权限,但是,假如中断没有如期返回,结果就不好说了。

这个效果正是溢出攻击程序希望做到并利用的。

图 15.6(a)所示为一段在 UNIX 下执行一个新 shell 的 C 代码。对该程序进行如下技术处理。

(1) 如图 15.6(b)所示,使用汇编工具转换为汇编语言代码,头尾加入技巧性的代码(阴影部分),利用 call 将下一条指令(本例中实际为字符串)的地址作为返回地址入栈,达到 pop 操作(第二行)后 esi 寄存器引用命令行字符串的目的。

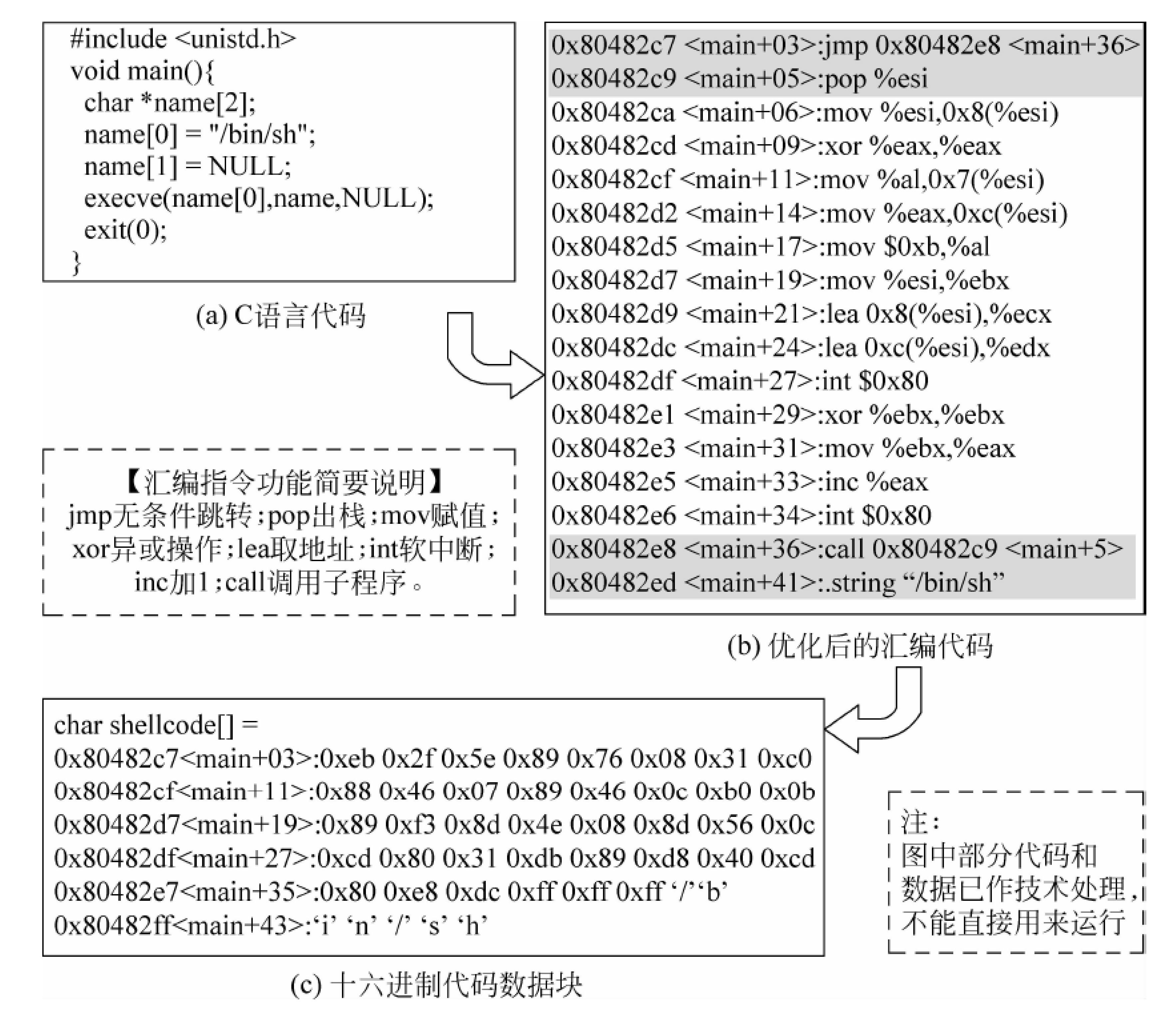


图 15.6　执行新 shell 的代码变换

(2) int 0x80 是操作系统提供的软中断调用,以寄存器 al 作为功能指示器,其他有关寄存器为输入参数。本例中使用了 0x80 软中断的两个功能: al=0xb 执行命令行,运行"/bin/sh"指代的 shell 命令; al=0x1 为退出操作。

(3) 为避免代码中出现数值 0,运用 xor eax,eax 替代相关赋值指令的技巧,防止以后使用 strcpy 函数的字符串复制操作中止。

(4) 最后获得如图 15.6(c)所示的十六进制程序代码数据块(数组)shellcode[ ],为下一步操作做好了准备。

缓冲区溢出攻击的思路是:使用字符串复制的系统函数 strcpy,把精简的 shellcode 代

码串作为“普通数据”植入程序堆栈区，同时修改返回地址使之指向植入后的 shellcode 起始位置，这样，具有 root 权限的 strcpy 即将执行完毕时，还来不及恢复低级权限，就被劫持“返回”去执行 shellcode 了。

为此需要设计一个同样巧妙的溢出攻击程序 overflow.c（如图 15.7 所示）作为母程序，以上一步得到的 shellcode 为数据，如图 15.7(b)所示，在通过 strcpy 复制函数搬移时，数据块后部的“返回地址”就会顺势覆盖原返回地址，一举多得地达到目标。

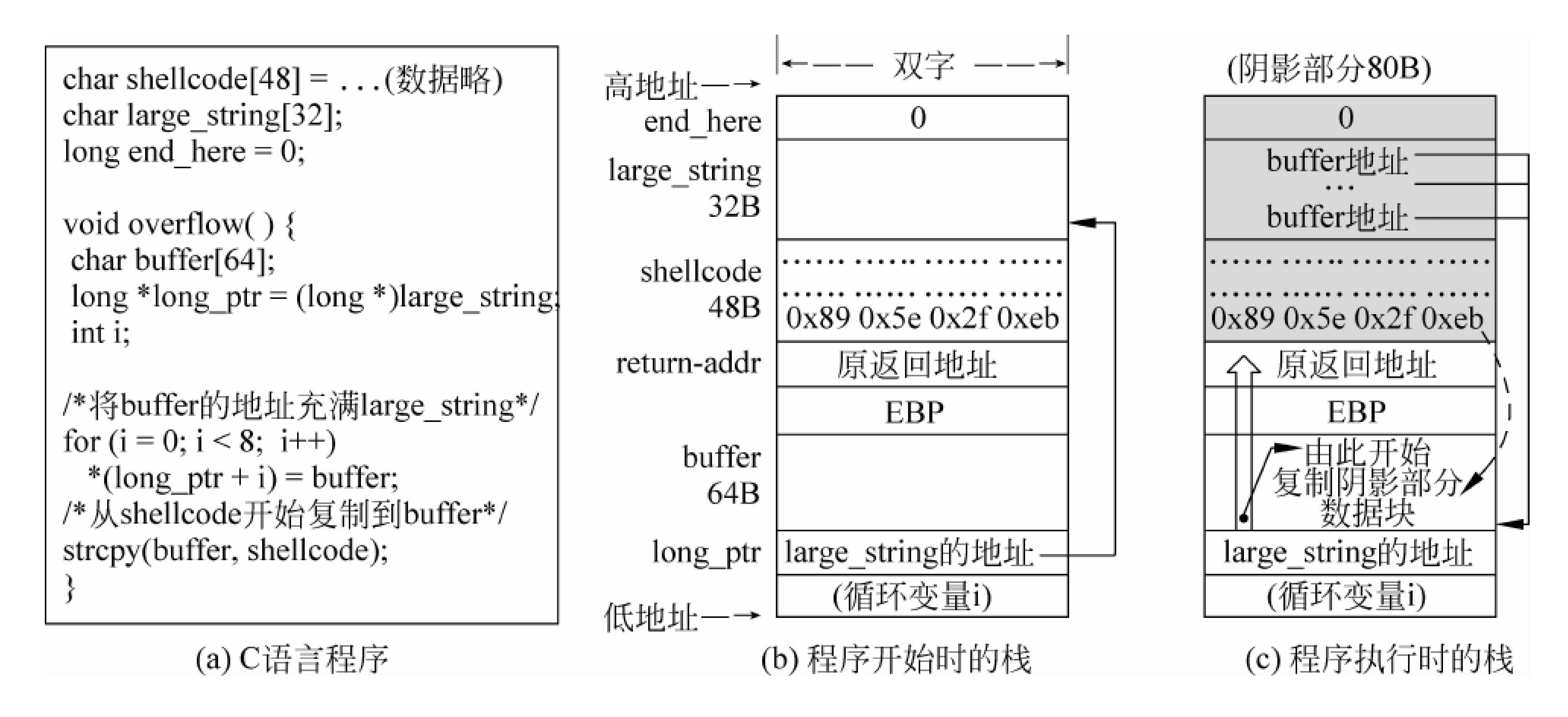


图 15.7　溢出攻击程序及栈内数据变化

细节决定成败。要达到攻击目的，还需要注意几个技术细节的处理：可通过在代码的合适位置插入单字节 nop 指令或在尾部添加非零数据的办法，使原 45B 的 shellcode 对齐双字长的整数倍；buffer 长度的选择依据是能够容纳 shellcode 所有数据；large_string 长度的选择依据是最后一步 strcpy 的复制内容足够延伸到 return-addr 区域。

实例中的 shellcode 和返回地址的数据块搬移是一种比较理想化的状态，前提是对栈内的数据结构（尤其是保存返回地址的位置）非常清晰，但实际情况较为复杂：不确定 buffer 的起始地址；buffer 的空间太小不足以容纳 shellcode（使 shellcode 可能覆盖原返回地址）等。针对前一种情况，如图 15.8(a)所示，采用在 shellcode 前插入一连串 nop（空操作）指令的方法，扩大猜测范围，增加对起始地址的命中率；后一种情况的解决方案如图 15.8(b)所示，只需简单交换 large_string 和 shellcode 的位置即可（图中 N 表示 nop，A 表示猜测的 shellcode 起始地址）。实际执行时需要多次尝试，不断调整，才能幸运命中。

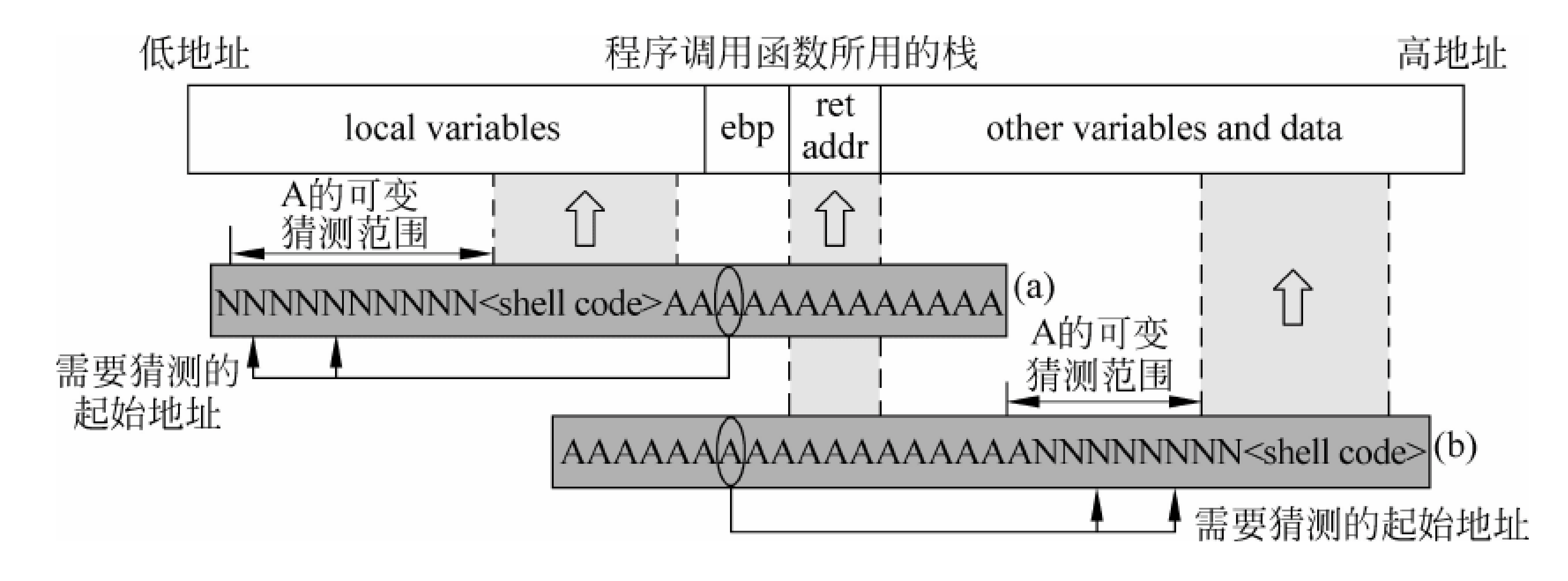


图 15.8　溢出攻击代码植入方案

15.2 注入攻击

攻击者在网络应用系统用户访问界面上输入事先严密设计的内容(恶意代码),并达到攻击目的,称为**注入攻击**(Injection Attack)。注入攻击往往针对 SQL(Structured Query Language,结构化查询语言)数据库存取操作,因此常称为 SQL 注入攻击。

数据库是网络信息系统的核心。用户列表、机密数据、内部文档、关键代码、系统配置、运行记录等,均存储在数据库系统中,并不断更新和积累。如果数据库系统受到攻击,不论是引起数据泄露,还是数据遭到篡改、伪造、删除,其影响必然是最致命的。

数据库访问采用 SQL 语句实现数据表单、数据字段的存取操作。不幸的是,编程人员惯常信手拈来的 SQL 语句,看似天衣无缝、完美无瑕,但在网络环境中却可能被人利用,变成系统安全的软肋。

例如,有以下常规的脚本语句。此脚本实现的功能是通过拼合程序设定的字符串和用户输入的字符串,生成一个有效的 SQL 查询。

```
var CityName;
CityName = Request.form ("CityName");
var sql = "select * from OrdersTable where CityName = ' " + CityName + " ' ";
```

该程序运行时,提示用户输入"CityName"。如果用户如编程者所愿输入字符串 Shanghai,则查询过程将完全按正常逻辑进行,由如下 SQL 语句(经拼合而来)执行完成,用户将从"订单表"(OrdersTable)中检索到与"Shanghai"相关的所有记录。

```
SELECT * FROM OrdersTable WHERE CityName = ' Shanghai '
```

但是,倘若有人不按常理出牌,输入了以下内容:

```
Shanghai'; drop table OrdersTable --
```

此时,程序将构造出以下 SQL 语句:

```
SELECT * FROM OrdersTable WHERE CityName = ' Shanghai';drop table OrdersTable -- '
```

分号(;)表示一个操作的结束和另一个操作的开始;双连字符(--)指示当前行余下的部分是注释内容,应该忽略。那么,当数据库处理该语句时,首先检索出 OrdersTable 中城市名为 Shanghai 的所有记录,随后,数据库将执行下一条 drop table 命令,删除 OrdersTable 数据表。攻击目的实现。

又如以下用以验证用户名和口令的 SQL 语句:

```
Select * from users where username = ' txtusername.Text ' and password = ' txtpassword.Text '
```

如果将 txtusername.Text 赋值(要求输入用户名时输入):

```
'or '1' = '1' --
```

将 txtpassword. Text 赋任意值(即输入任意口令,例如 0000),那么 SQL 脚本解释器中的语句就会变为:

```
Select * from users where username = '' or '1' = '1' -- and password = '0000'
```

显然,鉴于'1'='1'恒成立,用户验证总是得到通过。攻击目的实现。

可见,注入攻击是可行的,后果是严重的。

更严重的问题在于,只要注入的 SQL 代码语法正确,系统无法采用常规的编程方式来检测。

Web 访问中,常用形如 http://www. name. com/abc. asp? id=XX 等带有参数的 ASP 动态网页,参数有一个或多个,可为整型或字符串型,相关的 SQL 语句如 select * from 表名 where 字段=XX。那么,当攻击者在 XX 位置输入类似以上例子的恶意代码时,同样存在遭受注入攻击的危险。

软件设计中,差错(bugs)似乎是不可避免的,但差错有别于安全漏洞(或安全缺陷)。差错通常由笔误或逻辑错误所引起,理论上可以通过全面而严格的调试、测试来发现并解决。然而,安全漏洞一般很难捕捉,只有尽可能使设计更严谨、代码更规范,才能从根本上进行防范。比如:严格区分普通用户和管理员的访问权限;通过参数来传递变量,而非直接嵌入变量;对用户输入进行检查,一点发现有分号和双连字符等可疑字符即拒绝执行;运用数据库管理系统(Data Base Mangement System,DBMS)提供的安全手段;必要时采用漏洞扫描工具进行排查。

15.3　劫持攻击

中间人攻击(Middle-man)是一种劫持攻击手段,就像商品交易的中间人角色一样,隔离交易方,每一方传递的信息都被中间人截取,而双方都认为来自中间人的信息是另一方所发送的。中间人攻击通常结合其他手段综合运用。

在点对点通信方式下,中间人采用断开通信信道,将自身插入其间的方法,截留报文,并仿冒一方与另一方通信。

在网络多点通信方式下,中间人则采取更加多样化的手段,例如,分别骗取双方信任再进行仿冒通信(如利用公钥验证系统);伪造 IP 地址或 MAC 地址;扰乱并窃取路由;篡改 DNS 服务;破解 VPN 专网等。

黑洞(Blackhole)攻击主要存在于 Ad-hoc 网络中,属于另一种劫持攻击方法。

天文学家预言,在宇宙中存在一种特别的天体,其质量和密度无与伦比地巨大,由此产生超强的天体引力,以至于连光子都无法逃逸,所以必然是黑乎乎一团,即为黑洞。

黑洞攻击就是希望达到与天体黑洞同样的效果,使网络中所有结点的报文都发往恶意的黑洞结点。

黑洞结点首先接入 Ad-hoc 网络,运行与其他正常结点相同的链路层、网络层和路由协议。当黑洞结点与其他结点交换路由信息时,加入经过设计的可以影响路由选择的信息,这

些信息往往是子虚乌有的“可用信道”。其他结点当然信以为真,在路由信息中忠实记录了这些可选路径,建立起路由表。这样,当一个结点需要转发数据报文时,很可能就会发往黑洞结点。黑洞结点处理这些数据报文的唯一手段是丢弃。结果,Ad-hoc 网络中就像出现了一个天体黑洞,将各结点发送的数据报文纷纷吸走,有去无回。

Sybil 攻击与黑洞攻击有一定的相似性,但其方法主要是扮演不同结点的角色,如伪造或仿冒其他结点的 MAC 地址,以不同身份出现,达到干扰正常路由建立的目的。**Rushing 攻击**则采用抢先响应手段来阻止正常结点的通信。

虫洞攻击(Wormhole)也以干扰和劫持 Ad hoc 路由为目标。

虫洞是另一种宇宙学假定。通过虫洞可以弯曲或折叠时空,从一个点可以瞬间到达另一点,尽管两个点相距一万光年那么遥远。

虫洞攻击通常是由两个以上的恶意结点合谋,共同发动攻击。两个处于不同位置的虫洞结点相互协作,一个虫洞结点把收到的路由信息经由专用的信道传给另一个虫洞结点,另一个也把收到的路由信息直接传过来。如此一来,虽然两个虫洞结点相距甚远,却犹如穿越虫洞般只有一步之遥。如此经过两个虫洞结点掺假的路由信息,计算得到的跳跃数将有很大的机会比正常路径的跳跃数短,则数据报文就会乖乖地交给虫洞结点转发,结果当然很不妙。

CHAPTER 16

第16章 网络安全防范

网络安全防范(network security defence)是针对网络安全威胁,使用各种技术和管理手段,达到防止、发现、遏制、消除网络攻击的目的,保障网络与信息系统安全。

网络安全防范技术手段很多,各有其独特的作用,因此,在实际运用时需要注意几个方面的问题。

(1) 有的放矢地选择技术,在深入需求分析的基础上有所取舍。不做简单堆砌的泥瓦匠,要成为用心设计的建筑师。

(2) 一项技术解决一类问题,相互结合、相互补充才能形成完善的防范体系,不能有失偏颇,应该以全局的观点,注重全面防范效果提升。

(3) 讲究策略,讲究平衡,以最小的代价获得最佳防范效果。

(4) 不断跟踪网络安全威胁与防范的最新技术动态,不断调整和更新技术,才能保持长效的安全防范能力。

(5) 关注防范技术可能造成的负面影响,因为坚固的城堡可以更好地抵挡入侵,但同时也会在一定程度上禁锢自身。

网络安全方法策略和体系遵循水桶法则(短板效应),所以,技术措施应与非技术措施(包括物理安全、人员安全、安全管理等)紧密配合,不可或缺,不能厚此薄彼。特洛伊城不可谓不坚固,使强大的敌人无可奈何,却最终毁于一匹木马,教训是深刻的。

一些系统为了保障所谓彻底的安全,采用了完全物理隔离的办法,变成一个个信息孤岛,这种因噎废食的做法其实并不可取。网络的作用在于互连互通,不论是采用数据摆渡,还是人工端口切换,网络的优势都丧失殆尽了。

树欲静而风不止。我们应当认识到,安全防范技术再出色,也不是万无一失的。技术能力具有相对性。一方面,需要坚定信念,在战略上藐视网络黑暗势力,另一方面,要在战术上足够重视,未雨绸缪,让安全防范系统时刻处于活跃的、临战的、健康的、最佳的状态。

网络安全防范技术依照其基本原理可分为3种类型。

(1) **嵌入式安全防范**(Embedded Defence)指在信息交换路径上部署相

应的安全防范技术,可以是具有特定功能的设备(硬件),也可以是专门设计的协议、软件。嵌入式安全防范技术包括安全协议(或协议补丁)、虚拟专用网(VPN)、地址翻译(NAT)、访问代理(proxy)、网络防火墙(firewall)、病毒和木马查杀网关、垃圾邮件过滤等。

(2) **主动式安全防范**(Active Defence)指对网络信息系统的操作层面(用户)、信息层面(内容)、通信层面(组网)等关键环节加强网络安全防范措施,主动发现安全隐患,及时进行改进调整,防患于未然。主动式安全防范技术包括网络管理与系统管理、信息加密、数字证书、安全访问认证、虚拟子网(VLAN)、网络安全扫描与评估、软件安全修补等。

(3) **被动式安全防范**(Passive Defence)指两类安全防范措施:一类是通过部署的系统在网络安全威胁发生时能够及时发现、及时预警、及时采取措施,尽可能减少损失,防止灾难蔓延;另一类是通过对历史数据的分析,找出已经发生的攻击行为和事件,发现潜在的缺陷,以便采取针对性措施亡羊补牢。被动式安全防范不仅能够弥补嵌入式安全防范和主动式安全防范的不足和遗漏之处,而且具有动态检测的能力,是网络安全防范体系的重要组成部分。被动式安全防范技术包括入侵检测系统(IDS)、安全审计系统(SAS)、网页防篡改、实时监控系统、运行日志(System Log)等。

3 类网络安全防范技术之间的界限其实并不明显,是相互渗透、相互结合、相互协作的关系,共同满足保障网络和信息系统安全的需要。

16.1 嵌入式安全防范

16.1.1 防火墙

防火墙(firewall,FW)是网络安全防范体系的重要技术装备,属于嵌入式防范技术。如图 16.1 所示,防火墙作为一种网关(gateway)起到隔离内部网络和外部网络的作用,和路由器一起(经常合二为一)串联在内网出口位置。

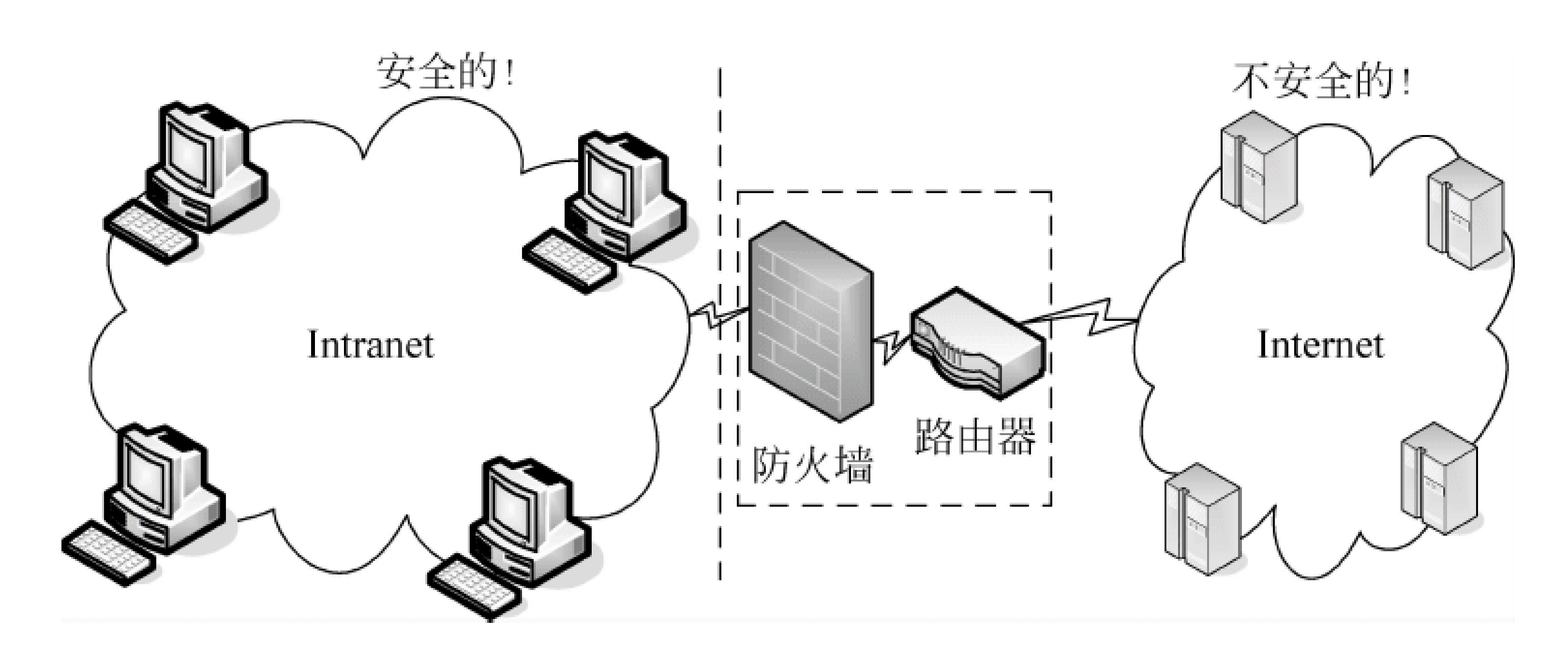


图 16.1 防火墙部署方式示意

防火墙的技术优势在于其透明性,即对内部网络的组网结构、计算机设备数量和分布、网络应用类型等均不敏感,有利于安装和使用。

防火墙的类型和功能很多,应该根据网络应用需求加以选择。从安全防范方法上看,防火墙有过滤(filtering)、代理(agent/proxy)等不同功能;从层次结构上看,防火墙有 3 层(IP)、4 层(TCP/UDP)之分;从防范对象上,有内网防火墙、病毒/木马防火墙、垃圾邮件防火墙;从技术实现上,分硬件防火墙、软件防火墙。从表 16.1 可以看出,不同的功能进行交

叉组合可以形成不同的防火墙功效。

表 16.1　防火墙功能

层次	过 滤 功 能	代 理 功 能
网络层	过滤 IP 地址、过滤广播、过滤探测报文、过滤无效报文	报文重组、NAT、防范 DoS 攻击
运输层	过滤 TCP/UDP 端口号	防范 SYN 攻击、防范其他 TCP/UDP 的 DoS 攻击
应用层	过滤内容	安全策略应用、病毒扫描、木马探测
用户层	过滤连接发起人	Cache、身份认证、计费

过滤技术和代理技术都是对协议报文、信息内容进行筛选，剔除不安全元素，放行安全的访问。但两者在介入深度上有所差别。过滤技术对每个通过的报文依次进行处理，是无状态的，报文间相互独立，互不影响，可以达到很高的处理速度。过滤技术可以比喻为日常生活中所用的筛子，分离尺寸大小不同的颗粒。代理技术则会参与协议会话过程，例如 TCP 连接过程、报文分割和拼接。因此代理操作是有状态的，同一个会话（或同一个流）的报文间相互有关联性。代理技术就像日常生活中的中介角色，处于每个交易的中间环节。

防火墙的过滤和代理操作具有不对称性，主要对来自外部网络（如 Internet）的信息流进行严格检查，而对内部网络发起的会话，检查力度相对较弱，这是由防火墙的防护职能所决定的。通常假定外部网络是不安全的，访问行为是不可信的，而内部网络较为安全、可控，需要受到保护。而且，大多数防火墙用于内网接入 Internet，绝大部分访问行为都是由内部计算机主动发起，从内到外的，很少有从外部网络主动访问内网主机（除非是网站系统），因此不对称的防护可以提高防火墙的运行效率，尽可能减少对正常网络访问活动的影响。

如图 16.2 所示，在 IP 层次的过滤检查方面，防火墙通过一系列的报文报头判别规则，使差错、虚假、伪造的 IP 报文无法通过，有针对性地防范了 Tear Drop、Fake IP、Ping of Death 等网络攻击活动。

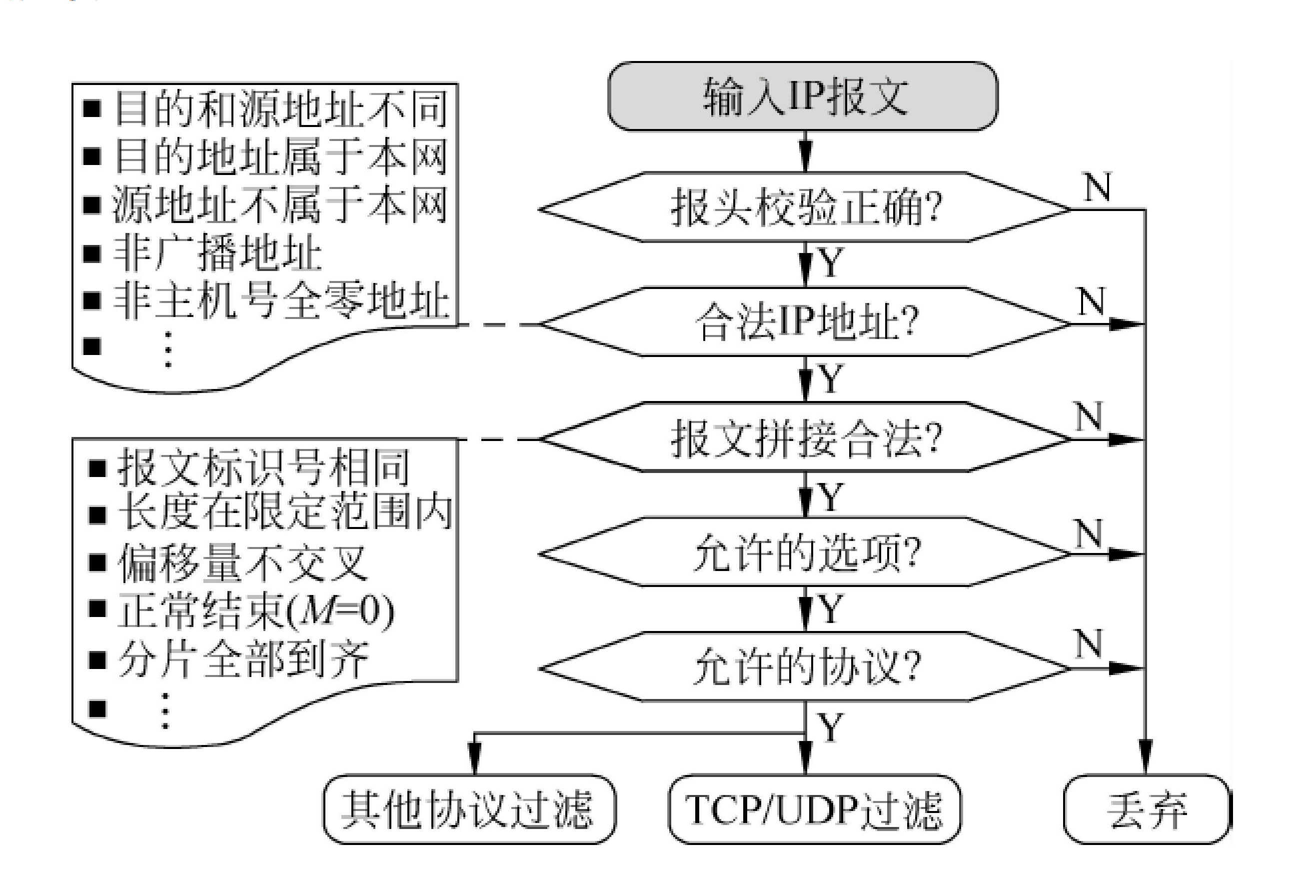


图 16.2　防火墙执行 IP 协议过滤流程

其中 IP 报文分片拼接合法性的检查可采用代理方式实现：由防火墙缓存需要拼接的 IP 报文分片（$M=1$），并根据规则检查每个分片。若发现不符要求的分组，则丢弃全部分

片；若收到最后分片($M=0$)并全部符合要求，则将缓存的报文全部转发到内网。在防火墙这一措施的限制下，来自外部网络的碎片攻击就无法穿透到内网主机。

再以 TCP 同步 DoS 攻击(TCP 3 次握手攻击)为例，说明防火墙代理机制所起的防范安全威胁作用。模拟算法流程如下。

(1) 外网攻击者发送 SYN，请求会话连接。

(2) 防火墙接收到 SYN，检查 IP 地址的有效性、源地址是否内网地址等，若判断为可疑的连接请求，丢弃报文(可不予以响应，以对抗探测)，结束，否则到下一步。

(3) 防火墙响应 SYN-ACK，启动超时计时器。防火墙只需匹配极少资源；计时器长度可以依据不同需求进行调整，适当缩短。

(4) 若计时器超时防火墙未收到 ACK，则释放该连接(不用发送拒绝报文，不占用带宽资源)，结束，否则到下一步。

(5) 防火墙若收到 ACK，立即向内网指定计算机(第(3)步已记录相关连接参数或已缓存原报文)发送 SYN，当接收到 SYN-ACK 后立即响应 ACK。TCP 连接建立成功。

防火墙在工作过程中使用**访问控制列表**(Access Control List，ACL)来灵活管理过滤机制。ACL 存放用于判别报文、连接与操作是否合法的相关参量，如 IP 地址、端口号等，可以动态维护、更新。使用 ACL 后，判别模块就成为一台过滤引擎，ACL 就是能量来源。

ACL 可以保存黑名单(black list；forbid list)，显性指出禁止访问的项目，如某个 IP 地址(或网段)、某个 TCP/UDP 端口号。访问控制引擎对黑名单采用负逻辑(negative logic)判别方式，即符合条件者不通过，否则通过。

ACL 还可以使用白名单(white list；permission list)，显性指出允许访问的项目。访问控制引擎对白名单采用正逻辑(positive logic)判别方式，即符合条件者通过，否则不通过。

黑名单制比较适合包含有限的禁止项目的情况，如明确需要拦截的不良网站。因为绝大多数网站是允许访问的，而且不胜枚举，则隐性允许访问。白名单制正好相反，适合包含有限的允许项目的情况，如仅允许 Web(端口 80)和 E-mail(端口 25 和 110)访问，其他应用(其他端口号)的报文都将被拦截。

灰名单(grey list)是一种颜色稍淡的特殊 ACL 黑名单机制，不同于黑名单的相对静态特性，灰名单可以由防火墙自动地、动态地进行调整，因此更为灵活，可称为临时黑名单。当防火墙监测到某个站点具有突发的不良表现(如正在发出蠕虫攻击，正在实施 DoS 攻击)需要进行过滤和拦截时，将其相关参数(如 IP 地址)自动移入防火墙的 ACL 灰名单，这样一来，只需简单判别 IP 地址就可以快速丢弃后续攻击报文，将可攻击行为的影响降到最低水平。如果经过一段规定时间(也可人工干预)后不再出现类似不安全问题，则可以将过滤地址从灰名单中释放，访问活动即恢复常态。例如，自动将正在发送垃圾邮件的 SMTP 服务器地址置入灰名单，可高效拦截潮涌式的大量垃圾邮件；当异常状况消失，自动将 SMTP 服务器地址从灰名单中去除。

防火墙主要用于保护内部网络的安全，但有时候也存在一些矛盾：内部网络上不同设备的安全需求是不同的，许多内部网络需要通过一些服务器系统(如 Web 服务器、E-mail 服务器、FTP 服务器)对外提供访问服务，这些服务器系统也需要一定的安全保护，但应避免复杂性，提高访问效率；同时，对外服务系统与内部应用系统间要有一定的互连关系，用于安全地交换数据。

如图 16.3 所示，可采用双防火墙的两层内部网络部署方案。两个防火墙分别采取不同的防范策略，以期在安全与效率间达到一定平衡。其中中间层以对外提供服务为目标，具有较强的开放性，更关注访问效率，而在安全上是可牺牲的缓冲地带，所以可以适当降低安全保护的强度，称为**非军事区**（De-Military Zone，DMZ）；内层则是真正的内部网络，受到最严格的安全保护。某些防火墙同时具有 DMZ 端口和内部网络端口，即是针对此目的而设计，使得用户不必分别添置两台防火墙，且部署和维护更便捷。

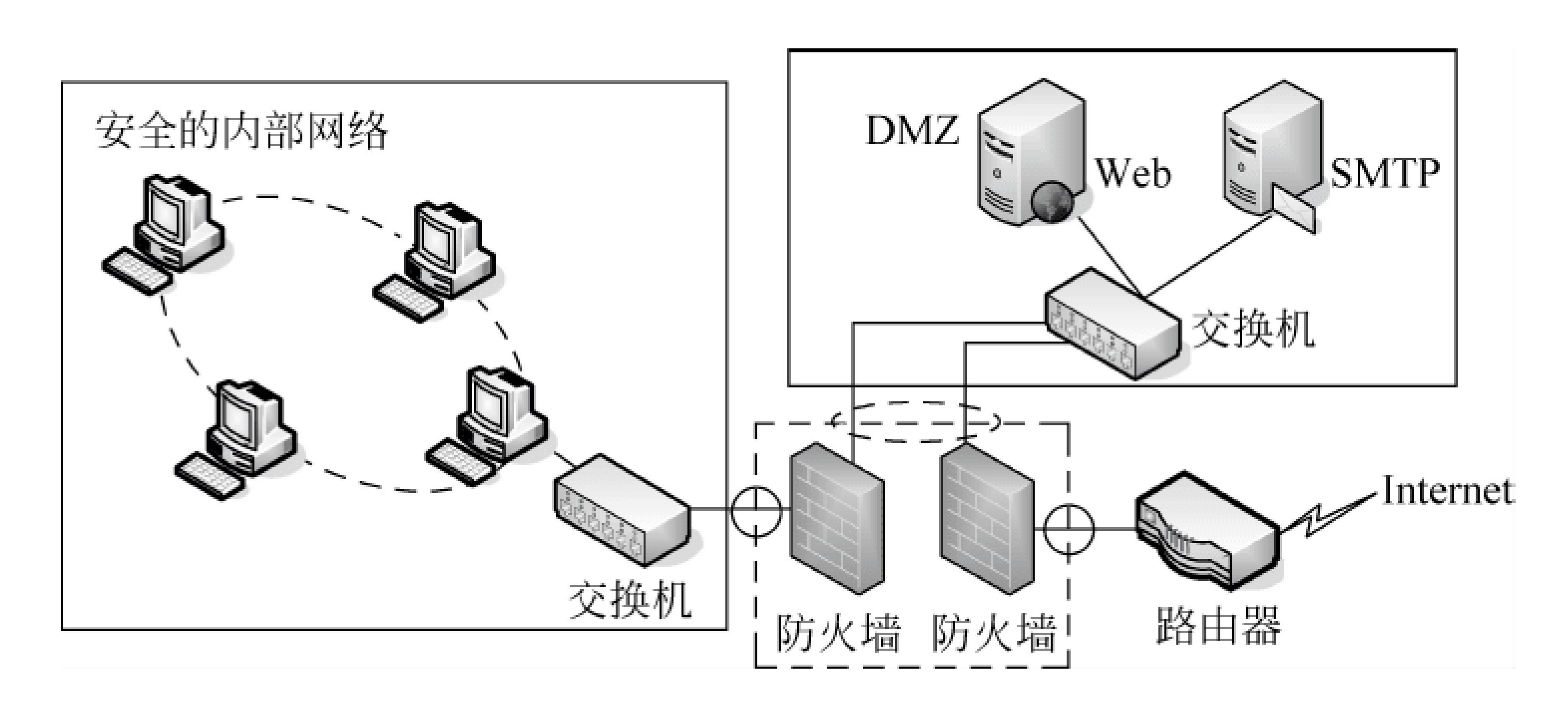


图 16.3　双防火墙和 DMZ 结构

防火墙运行时，应注意两个对立指标间的平衡：误报率（误检率）和漏报率（漏检率）。误报率（rate of false detection）指正常访问或数据误遭防火墙拦截，直接造成访问失败、数据（如邮件）丢失，误报率当然越低越好；漏报率（rate of missing）是检出率（rate of detection）的反面，指防火墙未能发现安全攻击的访问或数据，使漏网之鱼进入网络系统，漏报率也是越低越好。提高防火墙检查的严格级别，可降低漏报率、提高检出率，但有可能使误报率上升，反之，误报率降低，则可能出现漏报状况。不同的应用、不同的防火墙类型对这些指标的把握是不同的。例如，病毒防火墙应保证零漏报，否则一个漏网的病毒将使防线全面崩溃；垃圾邮件网关属于模糊判别的系统，就需要在相关参数间合理协调，使正常邮件不被拦截，垃圾邮件则越少越好，达到最佳防范效果。

16.1.2　代理

代理（proxy，agent）是计算机网络体系中的一种网关设备，用于协助网络用户完成访问任务，即用户将访问请求提交给代理，由代理完成对指定资源的访问，并把访问结果转交给用户。

可见，代理主要起到类似中间人的访问隔离作用，帮助网络用户完成其不能直接实施的任务。另外，代理可以设定缓存空间，存储网络访问的结果（如网页、文件等），以便向下一个相同的访问（可能来自不同用户）提供本地响应的、快速的服务，不占用珍贵的 Internet 接入带宽资源。代理采用缓存内容的定期更新和淘汰机制，以避免向用户提供过期信息。

正是由于代理可以对访问进行白名单式的控制，所以网络系统中的代理也是一种有效的网络安全设备。如图 16.4 所示，通过代理的设置，可以向指定的用户开放指定的应用、指定的 URL、指定的通信流量、指定的访问时段等，使用户的访问行为得到有效的约束，也限制了外部网络用户对内部网络用户的直接访问行为。

如图 16.5 所示，通过应用代理可以实现协议（包括端口号、IP 地址）的转换，用于穿越

防火墙的拦截(俗称翻墙)。工作原理是：内网客户机以防火墙允许的 HTTP 访问外网代理，代理正常访问相关应用服务，并将访问结果或回复以 HTTP 返回(例如采用隧道封装方式)，由客户机安装的代理插件转换为应用服务协议。

如果采用双代理技术，在内网安装另一台代理，与外网代理配合工作，客户机上就不需要安装特殊的代理插件，需要访问外网应用服务时以访问内网代理替代，适用性和灵活性更强。

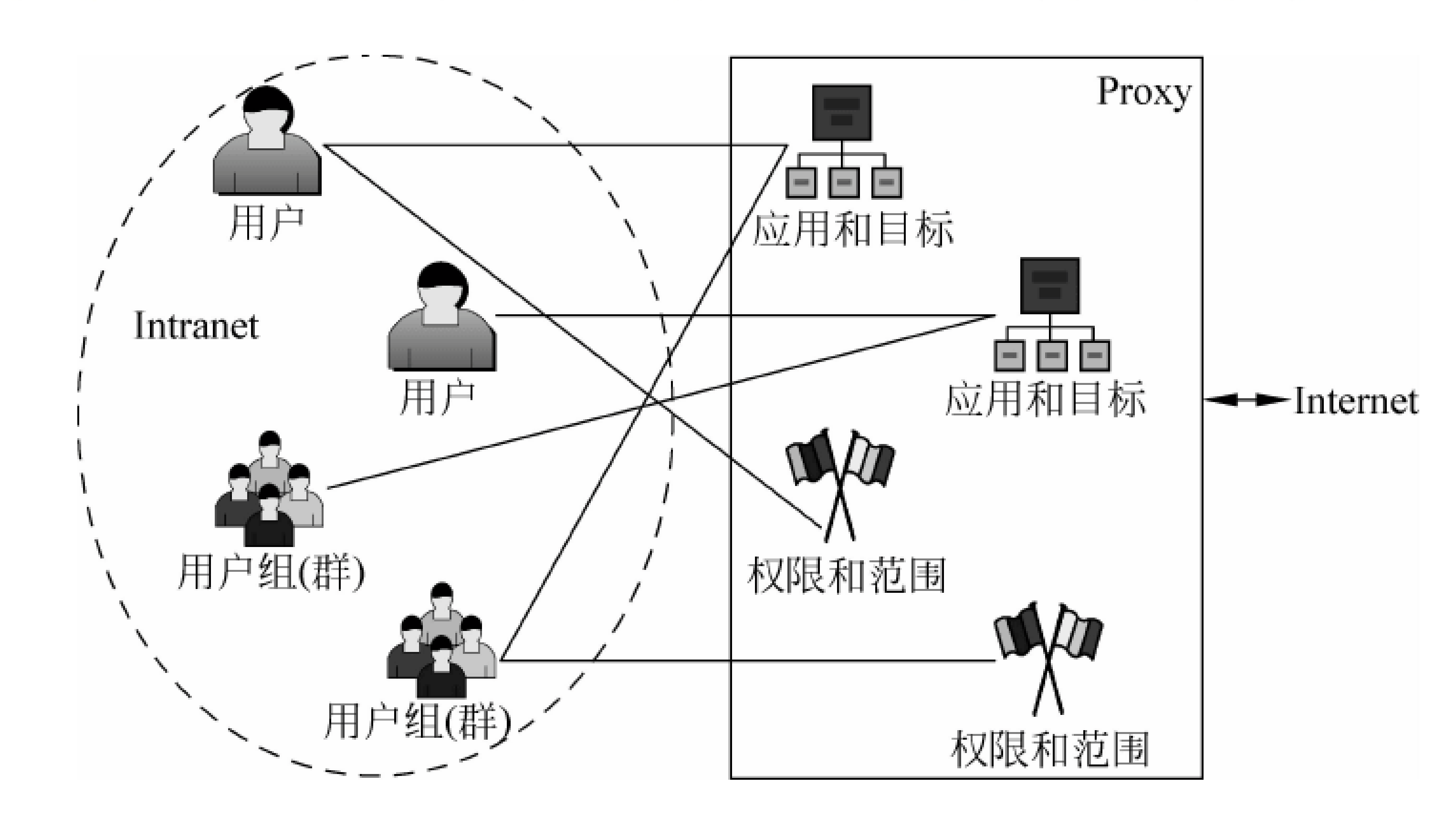


图 16.4　网络代理功能示意

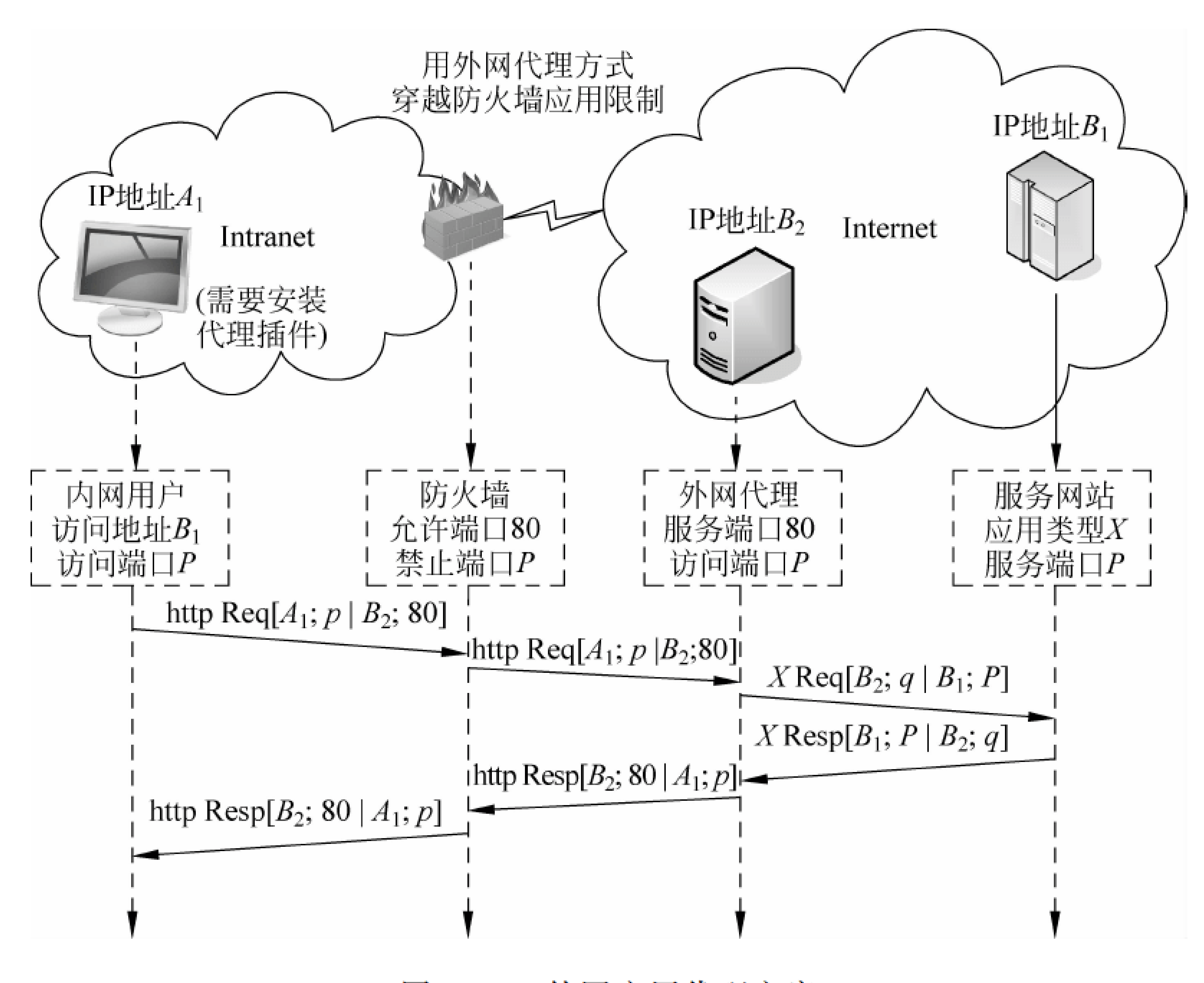


图 16.5　外网应用代理方案

16.2　主动式安全防范

16.2.1　安全口令

口令(password，pin)是一种方便而有效的网络安全防护措施。各种网络与信息系统的

用户身份认证都是依赖口令验证来完成。掌握了口令，就拥有了访问系统及其资源的权力。

口令一般和用户名(登录名、账号、账户名、ID 等)配合使用。口令技术可类比公钥体制。如果说用户名就是公钥，那么口令就是私钥。用户名可以公开(比如把 E-mail 地址印在名片上到处散发)，用于标识和区分不同用户的账户，而唯一防止账户被非授权者打开的手段就是口令。可见口令的安全性是极其重要的，对口令的保护也是信息安全的重中之重。

口令鉴别工作所承担的职能不仅是验证用户身份的合法性，还在于进一步对用户的访问进行必要的限定，以保护系统和数据的安全。用户认证系统一般通过角色(role)的定义来区分不同种类的用户，而不同的角色与事先指定的访问范围、访问方式、访问权限等相关联，达到用户认证→角色赋予→访问限定的目的。例如，超级用户(root)可以管理所有用户，访问所有资源，使用所有功能；系统管理员(admin)可以管理用户，但只能访问部分资源，使用部分功能；普通用户(user 或 guest)可以管理自己的账户，修改自己的口令，但只能使用指定范围的指定服务。

用户角色和权限的赋予可具有不同的细粒度，相应的系统复杂性和安全性也各不相同。安全管理越细致，安全性保障越好，系统复杂性和管理复杂性也越高。求大求全不是系统设计的目的。所以，针对不同的应用需求，应当在安全体系设计时找出最佳平衡点，优化系统配置。用户访问权限的设定可从粗到细划分如下。

(1) 可访问资源的 URL(或 IP 地址)。

(2) 可访问资源的协议类型(Web、E-mail、FTP 等)。

(3) 可访问的应用系统。

(4) 可访问的应用系统的功能(或时段)。

(5) 可访问的应用系统的功能操作方式(读/写、更改/添加/删除)。

(6) 可访问的应用系统的数据表。

(7) 可访问的应用系统的数据表的字段。

既然口令非常重要，为了防止被破解、猜测、窃听，理想的办法是采用**一次性口令**(One-Time Password，OTP)。一个口令(验证码)只使用一次，下一次一定发生变化，每次的鉴别口令都各不相同，可防范重放攻击。

如果每次都需要生成一个口令，口令很长，口令的字符组织无规律可循，等等，在原则上是正确的，但实际上行不通，因为没有兼顾用户如何获取口令，如何记忆口令，使用口令是否方便等因素。OTP 的任务在于解决实际使用口令和口令变化的矛盾。

如图 16.6 所示，OTP 的工作原理是：在用户登录过程中的口令传输时加入不确定因子，称为 salt，使用户口令不以固定的明文或密文方式在网络上传输，每次传送的验证码都不相同，而系统收到验证码后可以用相同的算法验证，达到一次性口令的目标。网络攻击者即使掌握了随机数、验证算法和 Hash 函数，但在不知道口令的情况下，只能采用暴力穷举破解的下策。

例如，登录验证码＝MD5(口令＋随机数)或 MD5(用户名＋口令＋随机数)，用户只需输入记忆的口令，系统自动拼接随机数后做 MD5 单向函数运算，生成每次不同的验证码。这里使用的随机数就是 OTP 的 salt，即干扰因子。随机数可以由客户端产生，在发送验证码时一起传送给服务端；也可由服务端产生，在询问口令时传送给客户端(相当于挑战式询问)；干扰因子也可由当前时间等参数代替。

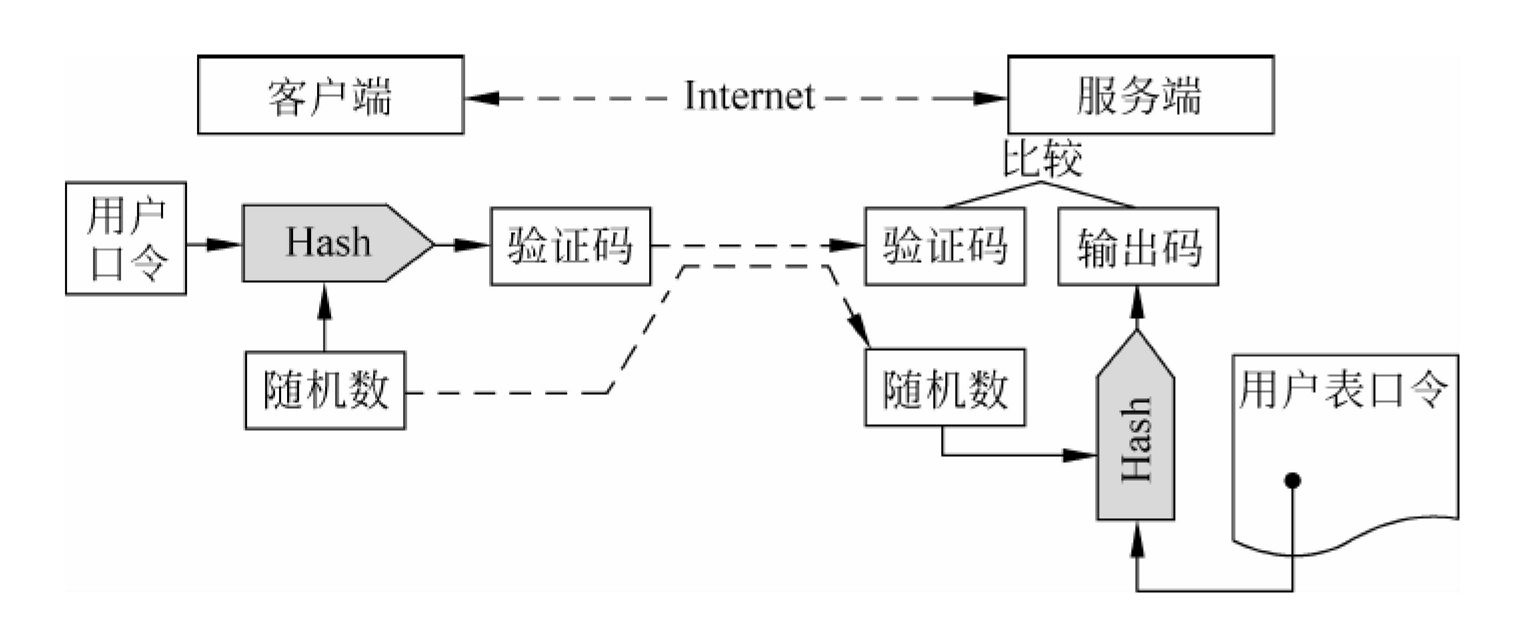


图 16.6　一次性口令原理

这种挑战/回答式一次性口令方法在 CHAP、Radius 等安全认证协议中得到运用。

配合使用特殊的设备或策略,也可实现其他一次性口令验证方案。

(1) 口令序列(S/KEY)。如图 16.7 所示,S/KEY 口令为一个单向的前后相关的序列,由 n 个口令组成。口令序列技术标准在 RFC 1760 中定义。客户端的口令序列是由用户口令和服务端提供的种子(seed)经 MD4 单向函数加密而成。客户端再通过连续 n 次单向函数(可采用 MD5、SHA-1 等算法),生成 n 个顺序排列的口令验证码序列。客户端登录时,逆向使用生成的序列。当用户第 i 次登录时,服务端用单向函数计算收到的验证码,并与上次保存的第 $i-1$ 个验证码比较,以判断用户的合法性。客户端按序使用口令序列,而服务端只需记录最新一次成功登录所用的验证码。但是,由于口令数是有限的,用户登录 n 次后必须重新初始化口令序列,且客户端和服务端应始终保持同步。

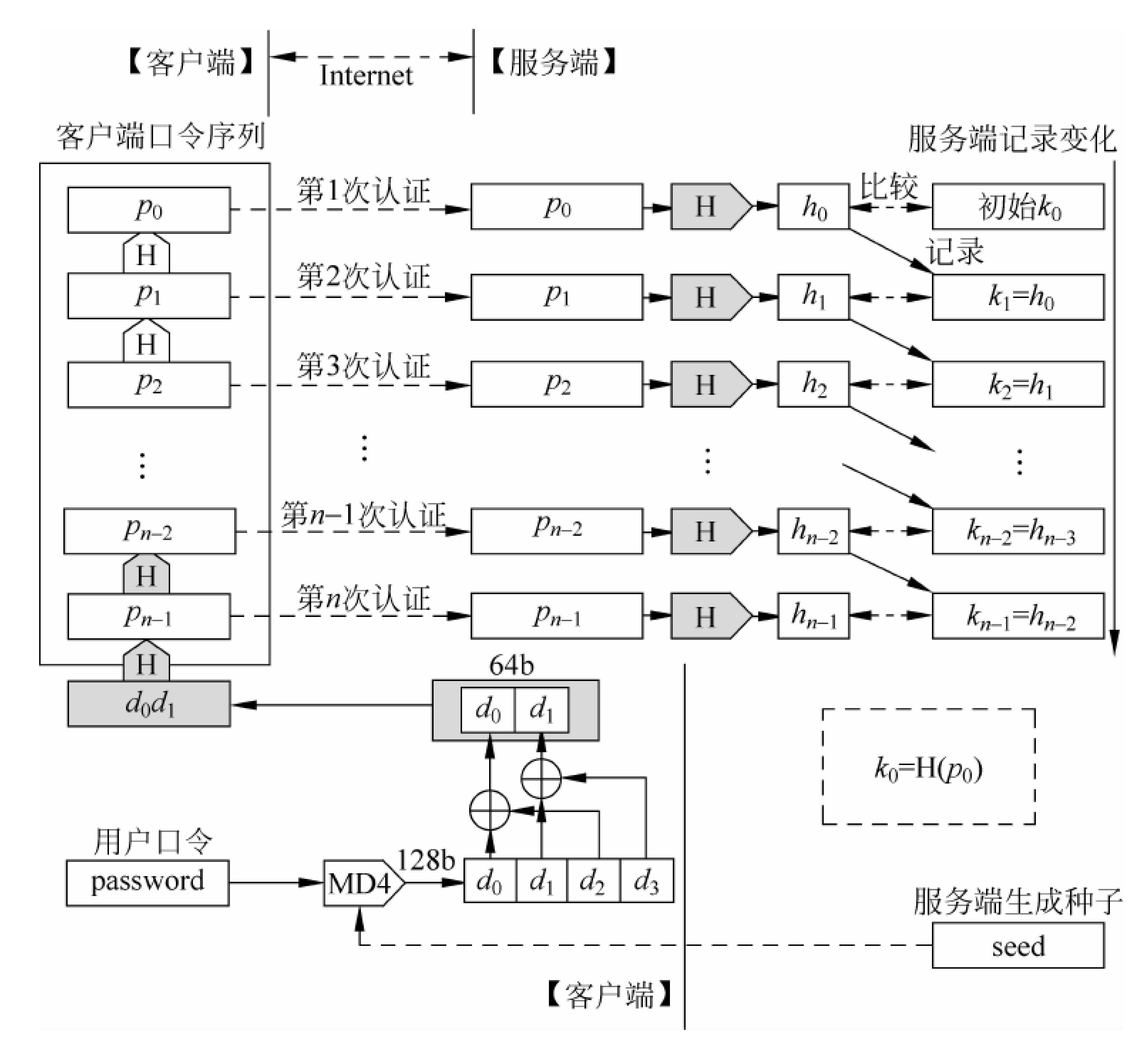


图 16.7　口令序列技术原理

（2）智能（令牌）卡（Token Card）。采用类似计算器的小卡片（或卡片式设备）计算一次性口令。如果采用挑战/回答方式，卡片配备有数字按键，便于输入挑战值；如果采用时间同步方式，该卡片每隔一段时间就会重新计算口令。有些卡片还带有 PIN 码保护机制。

（3）软件令牌（Soft Token）。用软件代替硬件，但限定软件安装运行的计算机设备、用户登录的地点（IP 地址或 Mac 地址）等。软件本身也需要操作口令保护。

（4）IC 卡或 USB 棒。在 IC 卡或 USB 棒上存储用户的秘密信息，这样用户在登录时就不用记忆自己的秘密口令了。

为了方便用户登录并访问不同的网络信息系统，可采用统一认证（Uniform Authentication）和单点登录（Single Sign-on，SSO）技术。两者的共同点是用户只需要一个账号，即可登录不同的系统，不用记忆多个账号的用户名和口令。不同点是：统一认证方式下，用户每打开一个新系统，都需要重新进行认证，适合用于 Internet 上不同的网站，这些网站已签订协议支持统一认证；单点登录方式下，用户只需登录一次，即可在不同系统间切换，不需要重新认证，适合于同一网站的不同子系统（或频道）间使用。

16.2.2　VLAN

虚拟局域网（Virtual Local Area Network，VLAN）是从逻辑（协议）上对网络资源和网络用户按照一定原则进行的划分。把一个物理网络划分成多个小的逻辑网络，形成各自的广播域，即成为 VLAN。例如，几个部门共同使用一个中心交换机，各个部门可划归不同的 VLAN。另一方面，也可以将一组位于不同网段（子网）上的用户在逻辑上归属到一个 VLAN 内，在功能和操作上与 LAN 基本相同。

VLAN 标准由 IEEE 802.1q 定义。如图 16.8 所示，VLAN 协议包含 4B 的标签头：2B 的标签协议标识（Tag Protocol Identifier，TPID）和 2B 的标签控制信息（Tag Control Information，TCI），插在 MAC 帧的源地址字段和长度/类型字段之间。TPID 兼容 MAC 长度/类型字段，固定为 0x8100（数值大于 0x0600，因此表示类型，不是长度）；TCI 由 3b 优先级字段（如表 16.2 所示）、规范格式指示符 CFI（1b）和 VLAN 标识符 VID（12b）组成。

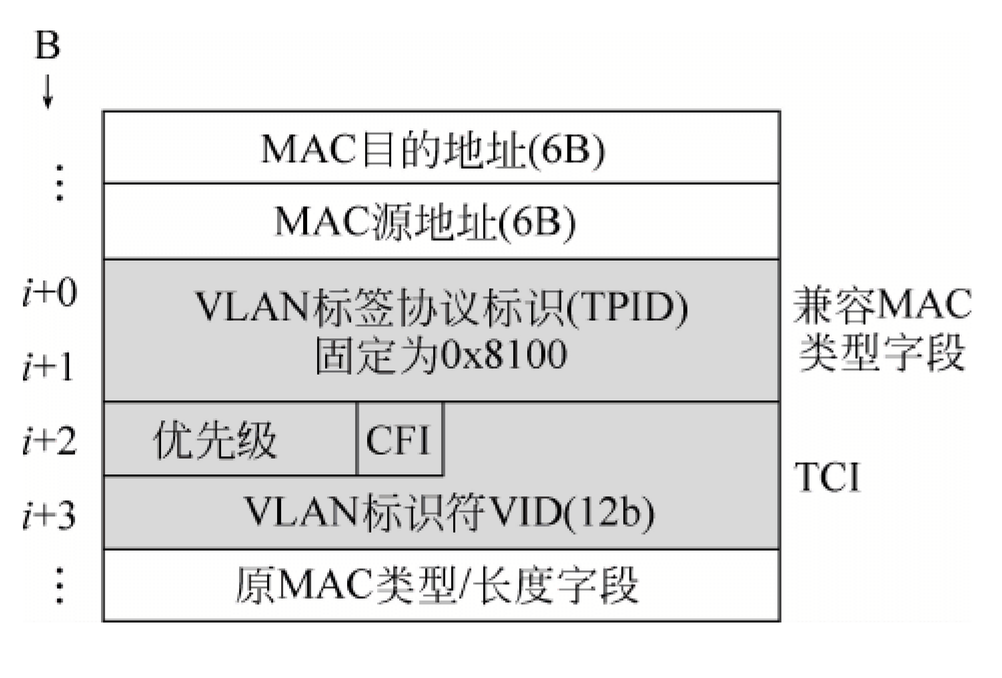


图 16.8　VLAN 报文格式

表 16.2　以太网 CoS 编码

业务类型	业务特征	CoS	协议举例
Network Control	适用于网络维护与管理报文的可靠传输，要求低丢包率	7	BGP、PIM、SNMP
Internetwork Control	适用于大型网络中区分于普通流量的网络协议控制报文，要求低丢包率和低时延	6	STP、OSPF、RIP
Voice	适用于语音业务，一般要求时延小于 10ms	5	SIP、MGCP
Video	适用于视频业务，一般要求时延小于 100ms	4	RTP
Critical Applications	适用于要求确保最小带宽的业务	3	NFS、RPC

续表

业务类型	业务特征	CoS	协议举例
Excellent Effort	比 Best Effort 的传输优先级稍高,用于一般的信息组织向最重要的客户发送信息	2	SQL
Best Effort	默认使用的业务类型,无优先发送的要求,只要求尽力而为的服务质量	1	HTTP、IM
Background	适用于不影响用户或关键应用的批量传输业务	0	FTP、SMTP

不同的 VLAN 间可制定各种访问限制,如不允许来自其他 VLAN 的访问,不允许离开 VLAN 的访问等。一个计算机(用户)可同时从属于不同 VLAN,需分别遵循相关 VLAN 的访问限定。

VLAN 有多种构造方式,可以根据环境条件以及应用的需要,有针对性地制定 VLAN 实施策略。

(1) 基于端口的 VLAN。根据以太网交换机的物理端口来划分 VLAN,可以跨交换机进行。优点是定义 VLAN 成员非常简单,只需将所有的端口都设定一遍;缺点是如果 VLAN 的某个用户离开了原来的端口,连接到了一个新的端口,就必须重新定义。

采用按不同地理位置(如不同楼层、不同大楼)的不同部门划分 VLAN,这种方式特别简单,因为往往各部门采用单独的交换机。

(2) 基于 MAC 地址的 VLAN。根据每个主机的 MAC 地址来划分 VLAN,所以也可称为基于用户的 VLAN。优点就是当用户物理位置移动时,即使移动到另一个交换机,VLAN 也不用重新配置;缺点是初始化时,所有的用户都必须进行配置,如果用户很多,配置的工作量非常大。此外,这种划分的方法也导致了交换机执行效率的降低,因为在每一个交换机的端口都可能存在很多个 VLAN 组的成员,这样就无法有效限制广播包。如果网卡可能经常更换,VLAN 就必须不停地更新。

(3) 基于协议的 VLAN。通过第二层报文中的协议字段,识别出上层运行的网络层协议,如 IP 或者是 IPX。当一个物理网络中存在多种第三层协议的时候,可采用这种 VLAN 的划分方法。但是现有的系统中一般仅有 IP,所以基于协议的 VLAN 很少有机会使用。

(4) 基于子网的 VLAN。根据报文中的 IP 地址决定报文属于哪个 VLAN,同一个 IP 子网的所有报文属于同一个 VLAN。优点是可以针对具体应用的服务来组织用户,用户可以在网络内部自由移动而不用重新配置自己的设备;缺点是效率较低,因为检查每一个报文的 IP 地址很耗时。同时由于一个端口也可能存在多个 VLAN 的成员,对广播报文也无法有效抑制。

与普通的 LAN 相比较,VLAN 具有一定技术优势,包括限制广播包,避免广播风暴;减少移动和改变的代价,可动态管理网络;实现虚拟工作组;子网间相互隔离,提高网络安全性;增强网络健壮性,将一些网络故障限制在局部范围内;降低维护成本。

16.2.3 VPN

虚拟专用网(Virtual Private Network,VPN)用于在不安全的网络环境(如 Internet)上构建安全的互连关系。

通过专线(电缆、光纤等)连接固然更有保障,但造价(或租用费)昂贵、灵活性差,大多数

情况下并非最佳选择。通过 Internet 进行互连，则灵活、廉价得多，极大地发挥了网络的互连互通优势。然而，Internet 毕竟存在许多安全隐患，严格地说是不安全的环境，这就需要通过一定的技术手段，在 Internet 基础上构筑安全的互连体系。

VPN 采用**安全隧道**(secure tunnel)跨越 Internet，采用安全协议来实现安全认证和数据加密，构造安全的数据传输通道，进而实现计算机设备或子网间的安全互连。

如图 16.9 所示，VPN 安全隧道贯穿 Internet，可安全传递两端计算机设备或子网间的信息，与 Internet 上其他计算机设备不产生通信关系，其他计算机也无法接入 VPN 或从 VPN 获取信息，好比一条穿越 Internet 大海的海底光缆。

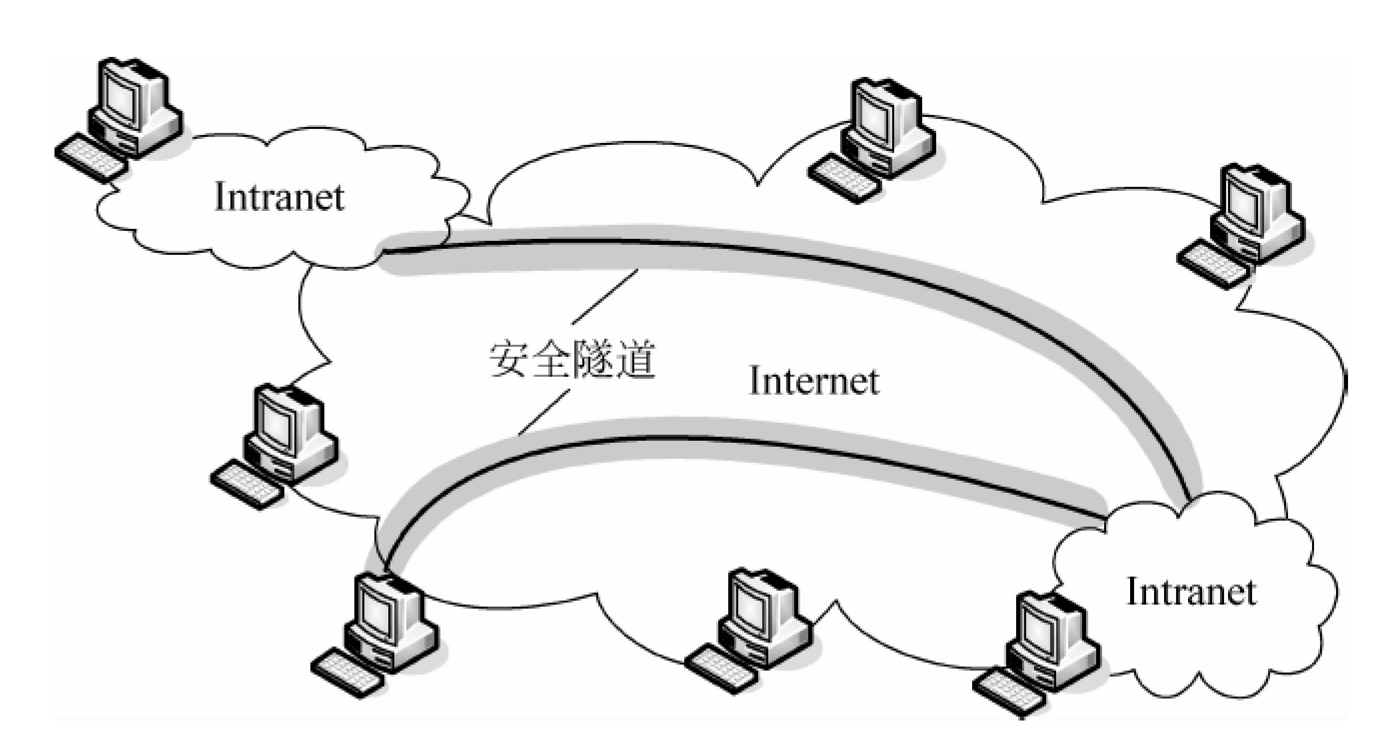


图 16.9　VPN 安全隧道示意

但是，VPN 安全隧道并非像海底光缆一样是固定的，而是随 Internet 路由而变。无论路由将报文转发到何处，总是位于安全隧道内。

企业内部网(Intranet)是一种采用 VPN 构造的安全网络，是企业(或机构)生产、办公、管理等经营活动的信息化基础设施。出于自身保护的考虑，企业网络需要采用良好的安全防范措施；出于网络互连和降低成本的考虑，采用 Internet 组网技术是最佳途径。如图 16.10 所示，结合安全性和开放性，采用 VPN 安全隧道技术构建的 Intranet 可实现单一地理位置的安全内部网络、多个地理位置的子网络互连成为统一的安全内部网络(如分公司、跨国公司)以及 Internet 终端的安全接入(如家庭办公、商务差旅)等多种需求。

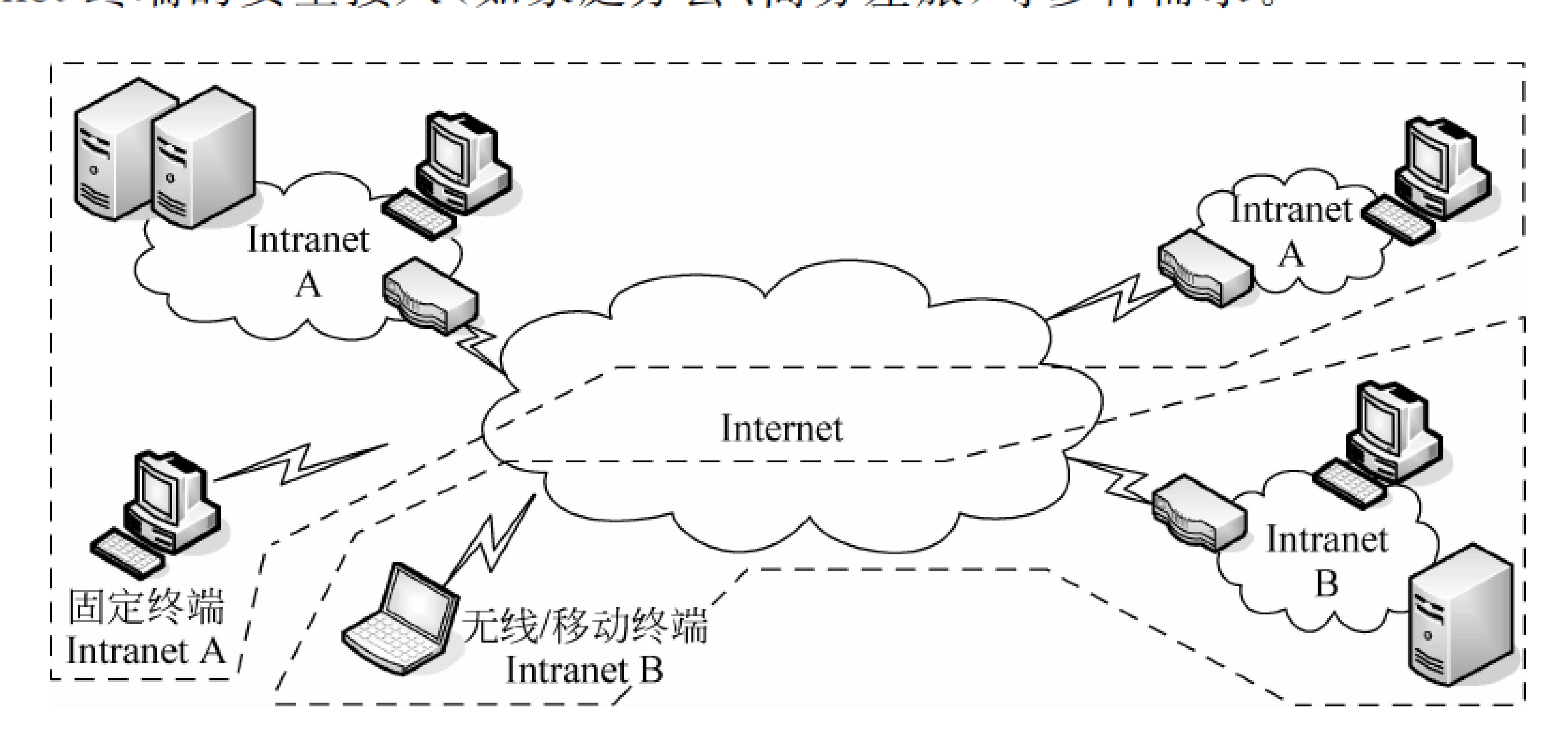


图 16.10　Intranet 组网示意

企业外部网(Extranet)用于满足企业(或机构)开展宣传、销售、合作等经营活动的另一类需要。如图 16.11 所示，Extranet 同样基于 Internet 并采用 VPN 安全隧道技术，但由于

不同企业 Intranet 不可能实现完全互连,为了满足业务互连的需求,采用协商确定的安全措施,通过 Internet 实现各自外连设备间的安全互连。依据合作关系的变动,Extranet 的成员可以动态调整。

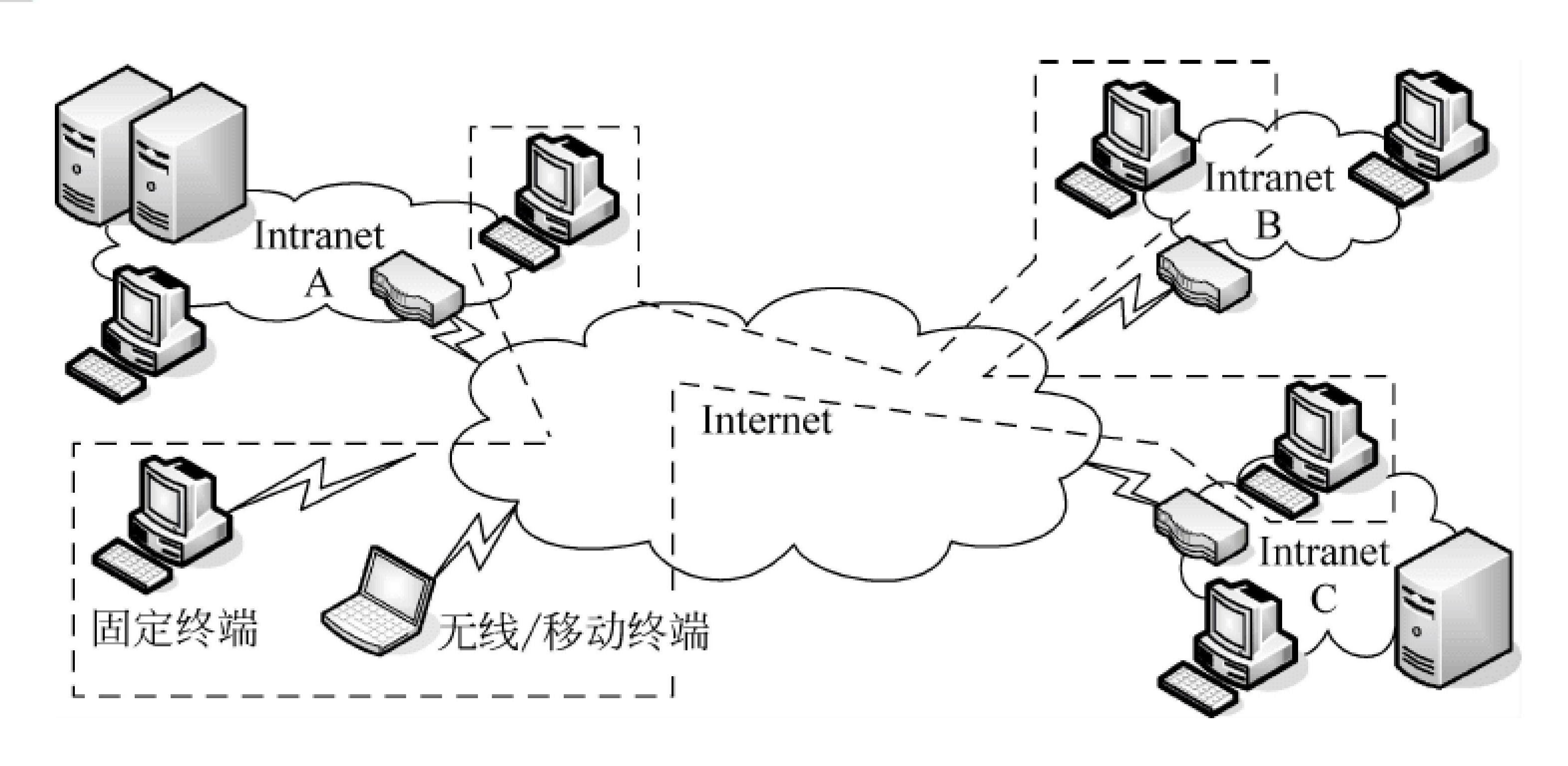


图 16.11　Extranet 组网示意

汽车公司需要国内外许多配件公司的合作,例如上游的引擎、轮胎、内饰、电子、钢铁、玻璃、石化,下游的广告、销售、维修、客服,各家企业都是独立企业,但为了保证配件供应、售前/售后服务准确配合“零库存”生产计划和销售计划,需要各个企业在生产、仓储、物流等方面紧密关联,并能适应合作方的各种变化。采用信息系统进行管理是必要的,Extranet 既能保证快捷、灵活的互连,又能保障信息安全。

适用于建立 VPN 安全隧道的安全协议有 PPTP、L2TP、IPSec 和 SSL 等。VPN 的构建除了采用基本的安全隧道技术外,应配合使用 PKI 等密钥管理体系。

VPN 在实际应用中有两种可行的组网方案。

(1) 网络透明方式:由用户自行配置 VPN 设备和系统,ISP 及 Internet 只提供普通的 IP 互连服务。

(2) 用户透明方式:由 ISP 提供 VPN 服务,用户端使用普通 IP 互连设备。

16.3　被动式安全防范

16.3.1　网页防篡改

Internet 网站经常会发生受到攻击、网页被篡改的情况(俗称网站被黑)。由于网站的安全防范体系存在缺陷,使 Web 服务器管理权限被接管,攻击者就可以长驱直入,远程修改 Web 服务器上存储的 HTML 文件,例如网站首页文件 index.html,然后随心所欲地修改任意部分,换成任何内容,达到篡改网站网页的目的。

引起网页被篡改的原因可能包括管理账号密码被攻破;服务器操作系统存在安全漏洞并且没有及时打补丁;应用系统遭到注入式攻击;系统被植入木马或其他恶意代码。

网页防篡改系统则专门设计用于防止网站网页被篡改行为,一旦发现文件有任何修改的迹象,立即予以恢复。因此,网页防篡改本质上是一种文件保护措施,但由于 Web 文件经常会更新,普通的、静态的文件保护无法满足需要。

如图 16.12 所示,网页防篡改系统部署于网站内容发布服务器与网站 Web 服务器(集

群)之间,也可作为内容发布服务器应用系统的一部分。

网页防篡改系统主要采用以下技术对 Web 服务器上的所有 HTML 文件和相关脚本、图片、视频等文件进行严格检查。

(1) 所有文件更新并发布时,采用数字签名技术;当文件因访问需要而被调用时,使用数字签名检查是否有改变。

(2) 监测用户 URL,防止注入式攻击。

(3) 定时轮询检查所有已发布的文件的完整性,并检查是否有未知的新增文件。

(4) 检测 Web 服务器运行情况,包括负载情况、带宽占用情况、活跃进程以及系统操作日志,以便及时发现异常状况。

如果发现任何文件有改动,立即使用原始备份进行替换(恢复),并进行重点监测,防止文件被重新修改,同时向系统管理员发出报警。如果修复无效,可采取断开相关服务器网络接口等措施。

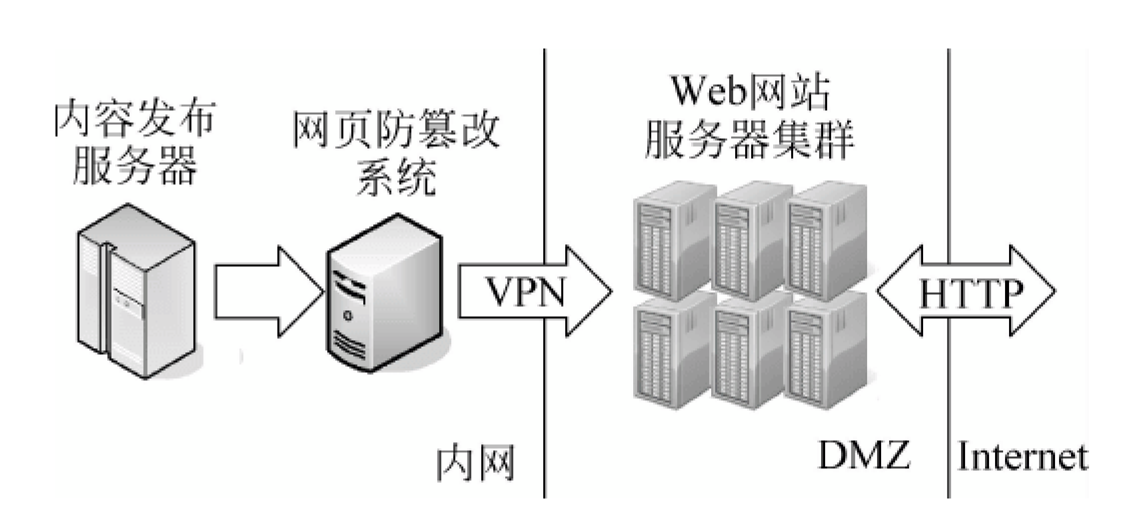


图 16.12　网页防篡改系统部署示意

16.3.2　入侵检测

入侵检测系统(Intrusion Detection System,IDS)是用于在内部网络上探测和发现网络入侵活动的系统,是动态安全防范的核心技术之一。如图 16.13 所示,相比操作系统加固技术、防火墙隔离技术等静态安全防御技术对网络环境下日新月异的攻击手段缺乏应变能力的缺陷,基于最新的可适应网络安全技术和 P2DR(Policy Protection Detection Response)安全模型的 IDS 可以深入地解析入侵事件、入侵手段及被入侵目标的漏洞等,及时发现和报告可能的入侵现象,并与其他网络安全设备联动,实施拦截。

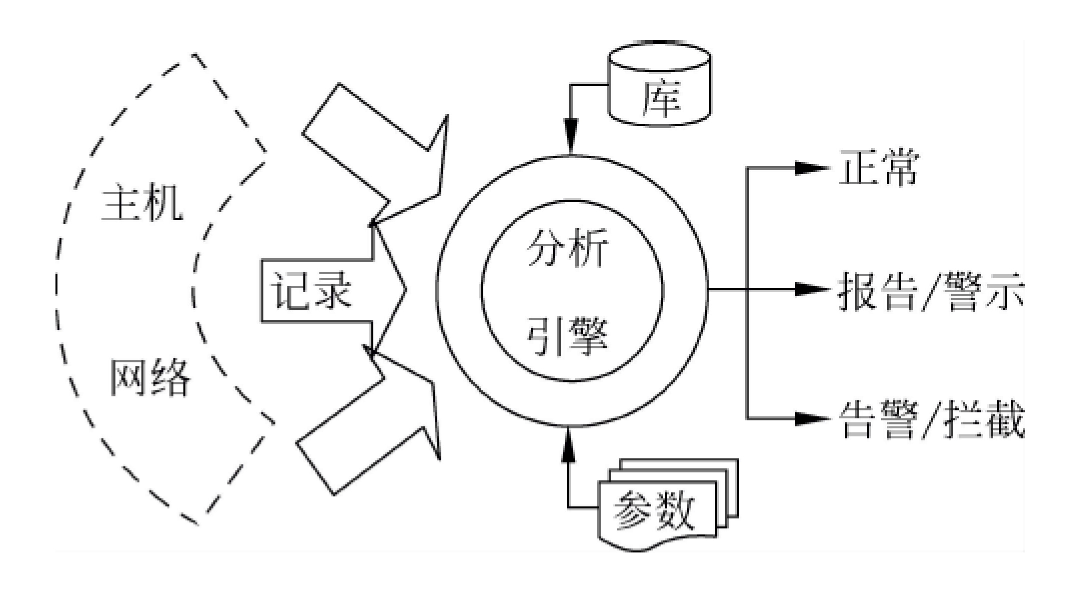


图 16.13　IDS 工作原理示意

IDS 采集信息的手段是在网络信息系统中部署软件或硬件探针(probe),对于不适合安装探针的设备,可采用智能代理(agent)。多台设备可合用一个探针。探针起到类似传感器

的作用,负责实时搜集各种操作信息,特别是网络访问的有关信息,如源与目的 IP 地址、时间、用户账号、操作序列、访问资源、流量等,汇总到 IDS 控制台(console)或管理中心,经过数据分析,可以得到网络访问的行为模式,并进行分析判定(如图 16.14 所示)。

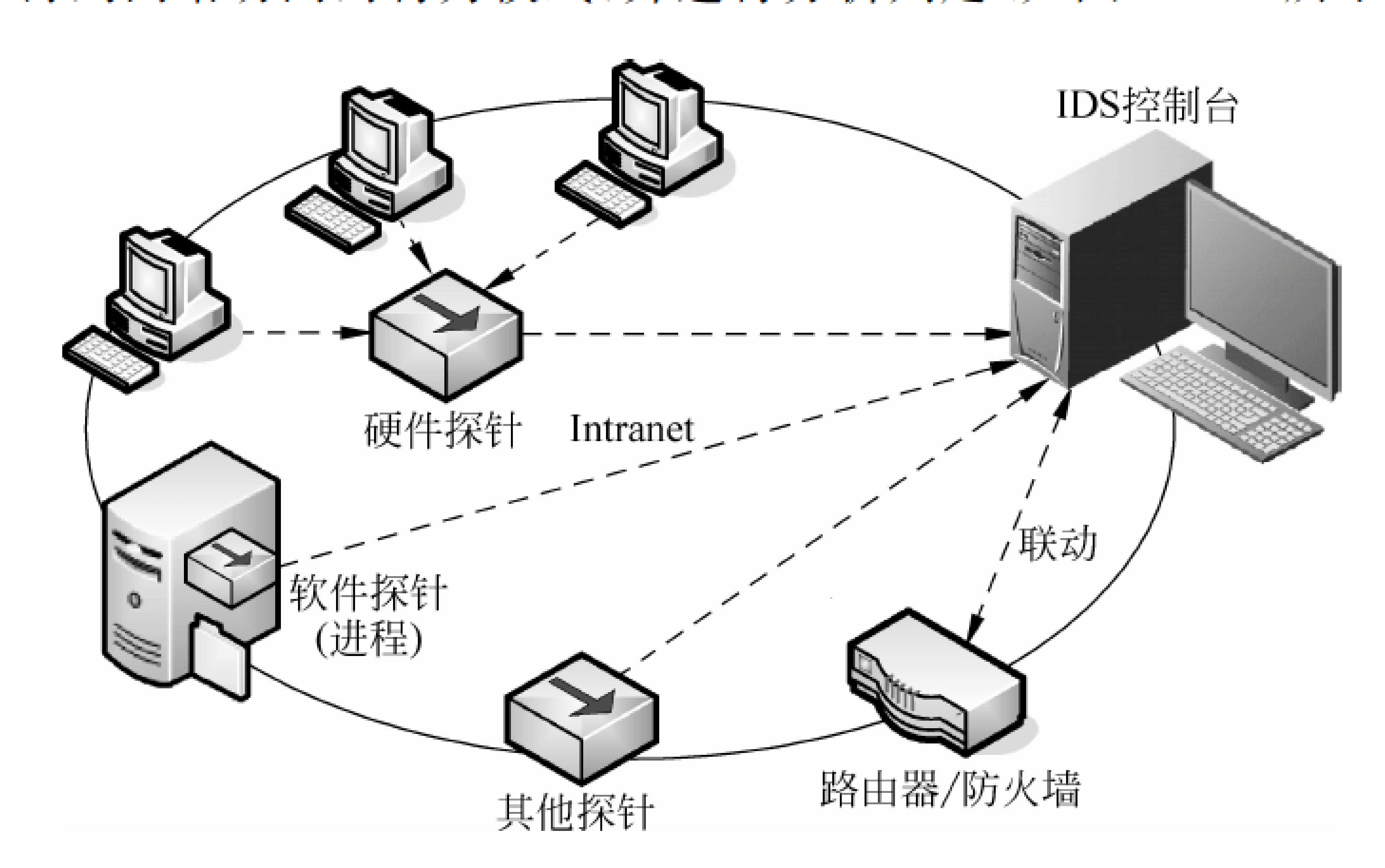


图 16.14 IDS 体系结构示意

如图 16.15 所示,IDS 系统由模式匹配机、系统剖析引擎、异常检测器、响应和恢复机制以及入侵模式库等模块组成,也可根据不同的技术类型选择部分模块。系统分析的结果被保存到数据库中随时备查。

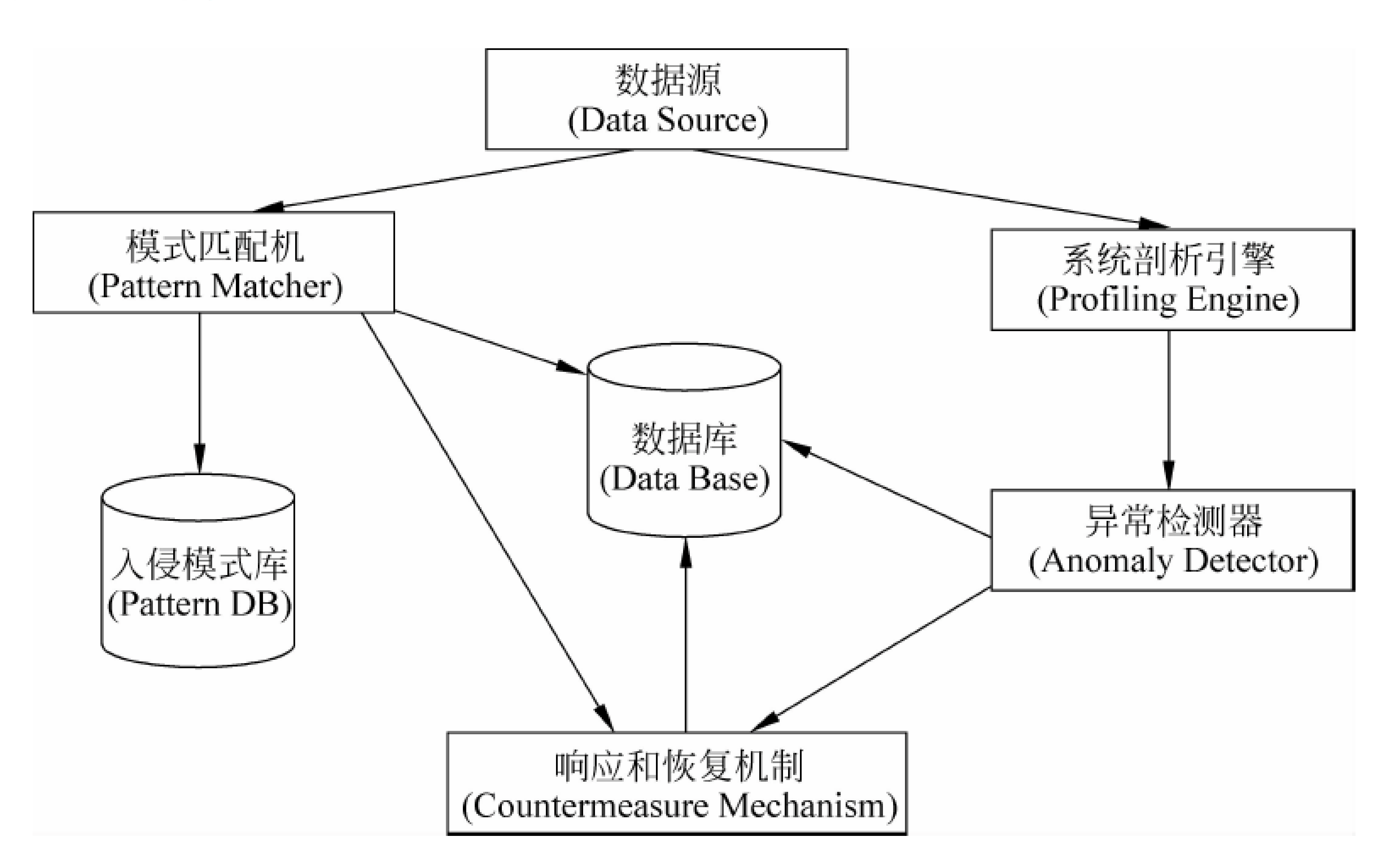


图 16.15 IDS 通用模型

IDS 主要研究入侵行为的过程与行为特征,从分析方式上主要分为两种技术类型。

(1) 模式发现技术。假定所有入侵行为和手段(及其变种)都能够表达为一种模式或特征,那么所有已知的入侵方法都可以用匹配的方法发现。模式发现的关键是如何表达入侵的模式,把真正的入侵与正常行为区分开来。模式发现的优点是误报少,局限是它只能发现已知的攻击,对未知的攻击无能为力。

(2) 异常发现技术。假定所有入侵行为都是与正常行为有差异的。如果建立系统正常行为的轨迹,那么理论上可以把所有与正常轨迹不同的系统状态视为可疑企图。对于异常

阈值与特征的选择是异常发现技术的关键。例如，通过流量统计分析将异常时间的异常网络流量视为可疑。异常发现技术的局限是并非所有的入侵都表现为异常，甚至可能被攻击者“训练”而规避。

实际应用的入侵检测系统主要以较为成熟和可靠的模式发现技术为主，适当运用异常发现技术。

IDS 从实现方式上分为两种：基于主机的 IDS 和基于网络的 IDS。一个完备的 IDS 一定是基于主机和基于网络两种方式兼备的分布式系统。另外，能够识别的入侵手段的数量多少、最新入侵手段的(模式)更新是否及时等都是评价 IDS 性能的关键指标。

新一代的**入侵防范系统**(Intrusion Preventing System，IPS)是从 IDS 的基础上发展而来，更强调主动介入式的安全保护。

16.3.3　安全审计

安全审计(Security Audit，SA)系统与 IDS 在防范功能、技术原理、系统结构等方面都非常相似，但 SA 主要从事网络信息的静态、事后的分析工作，正因为如此，SA 具有独特的优势，可以进行海量信息的分析，数据分析也能更为深入和细致。此外，SA 还能够进行回顾性分析，使用最新的分析模型对原有技术条件下曾判别为干净的记录重新进行安全隐患挖掘。

SA 可以对日志、系统文件等大量数据进行分析，除了采用模式匹配方法外，还可以采用数理统计分析、数据完整性分析等技术手段。

统计分析方法首先给系统对象(如用户、文件、目录和设备等)创建一个统计描述，统计正常使用时的一些测量属性(如访问次数、操作失败次数和延时等)。测量属性的平均值将被用来与网络、系统的行为进行比较，任何观察值在正常值范围之外时，就认为有入侵发生。具体的统计分析方法有基于专家系统的、基于模型推理的和基于神经网络的分析方法。

完整性分析主要关注某个文件或对象是否被更改，包括文件和目录的内容及属性，在发现被更改的、被木马化的、被病毒感染的应用程序方面特别有效。完整性分析利用数字摘要技术，能识别哪怕是微小的变化。其优点是不管模式匹配方法和统计分析方法能否发现入侵，只要是成功的攻击导致了文件或其他对象的任何改变，都能够被发现。可以在每一天的某个特定时间内开启完整性分析模块，对网络系统进行全面扫描检查。

在 IDS 和 SA 系统应用中，经常部署攻击陷阱(attacking trap)，以诱使网络安全攻击进入预设的目标，既起到转移注意力的作用，防止真正的数据被窃取或破坏，又可以通过特别预设的监控程序及时发现攻击行为，并进一步跟踪和寻找攻击源。例如，部署看上去比其他主机更像核心服务器的设备，开设具有通用的管理员账户名称的虚假管理员账户，存放显得很有价值的杜撰用户信息列表文件或数据库等。相对于网络安全威胁中常用的陷阱攻击手法，攻击陷阱策略可以看做是以其人之道还治其人之身的做法，符合兵不厌诈的用兵之道。

CHAPTER 17

第 17 章　网络冗余技术

17.1　冗余技术原理

冗余(redundancy)技术是一项有趣的技术。冗余不是指多余、重复、浪费或可有可无,但它确实需要依赖一个或多个副本。当然,冗余并非复制自身那么简单,重点是具备后续的、可恢复的手段。

冗余是必要的。硬件的失效、软件的错误、设备的老化、线路的中断,都有可能导致连接的失败、数据的丢失。只有一定的冗余,才有可能在故障、灾难发生时或事后进行补偿和复原。例如,写这本书时,需要经常复制文档,虽然占用了一定的存储空间,但不再担心文件损坏,因为这是经常发生的事情。而且,最好把副本保存在优盘里,防止电脑损坏而全部工作泡汤,否则就是噩梦了。此外,冗余也是扩展的需要。一个好的设计需要考虑适当的余地,就是发展空间,这样才能保证在需求扩张的时候从容应对。

一个优质服务的大饭店的客房不能总是 100% 被占满,否则就无法满足一些突然出现的 VIP 贵宾的要求,而惹怒常客绝对是一件蠢事。

建筑设计中,应力的计算可以很精确,但最后的设计图纸上总是保留很大的保险系数。相信没人愿意呆在一幢受力处于临界值的楼房中。

远洋船只上带的食物和水也要有较多的宽余度,天有不测风云,谁也无法预料意外情况,保障安全和健康是第一位的。谁能保证船到码头刚好吃完最后一个罐头,喝光最后一滴水?

另一个好的例子是:911 灾难发生后,有一家在世贸中心大厦有办公机构的公司很快就恢复了所有客户的资料,并以最快的时间全面投入正常运作。这就是冗余(备份)的威力。

备份(backup)是一种容灾和恢复(disaster recovery)措施。在人为的(人祸)或不可抗拒的灾害(天灾)发生时能够尽快复原,避免更大的损失,防止灾难蔓延。

网络与信息系统需要保障数据传输的完整性、准确性、可靠性、安全性和

高效性，因此，系统设计的各个方面经常运用冗余技术。本章主要讨论路径冗余、设施冗余、存储冗余和数据冗余四种类型。

思考：计算机系统中的 Cache 是否属于冗余技术？

17.2　路径冗余

路径泛指计算机网络中数据的传递通路。在物理上，数据传输的路径为传输媒介，或称线路；在逻辑上，数据传输的路径就是路由。

17.2.1　线路冗余

线路冗余有多种策略，其目的都是当其中一条线路发生故障（如中断）而不能提供传输服务时，由冗余（备份）的线路来承担通信任务。

除了考虑网络布线时的冗余度，为网络扩容和技术升级做准备，线路备份应优选不同走向的线路作为相互备份，才能保障备份的有效性。这是由线路发生故障的特点所决定的，例如一同被挖断，一同被盗窃，一同老化失效，一同受外界干扰等。

备份一般有**冷备**（Cold Backup）和**热备**（Hot Backup）之分。冷备是指备份线路平时并不工作（参与通信），只有当主线路失效时，才启动使用备份线路；热备是指主、备（或互为主备）线路一起工作，任意一条线路失效都不影响通信的正常进行（但通信能力可能降低）。

实际上，热备也可以采取不同策略：第一种是主、备线路共同承担通信流量，起到**负载均衡**（Load Balance）的效果；第二种是主线路工作，而备线路上并不传输数据，除非主线路发生故障（注意与冷备方式的区别）；第三种是两条线路分别为不同的子网工作，当其中一条线路故障时，另一条线路同时为两个子网服务（如图 17.1 所示）。

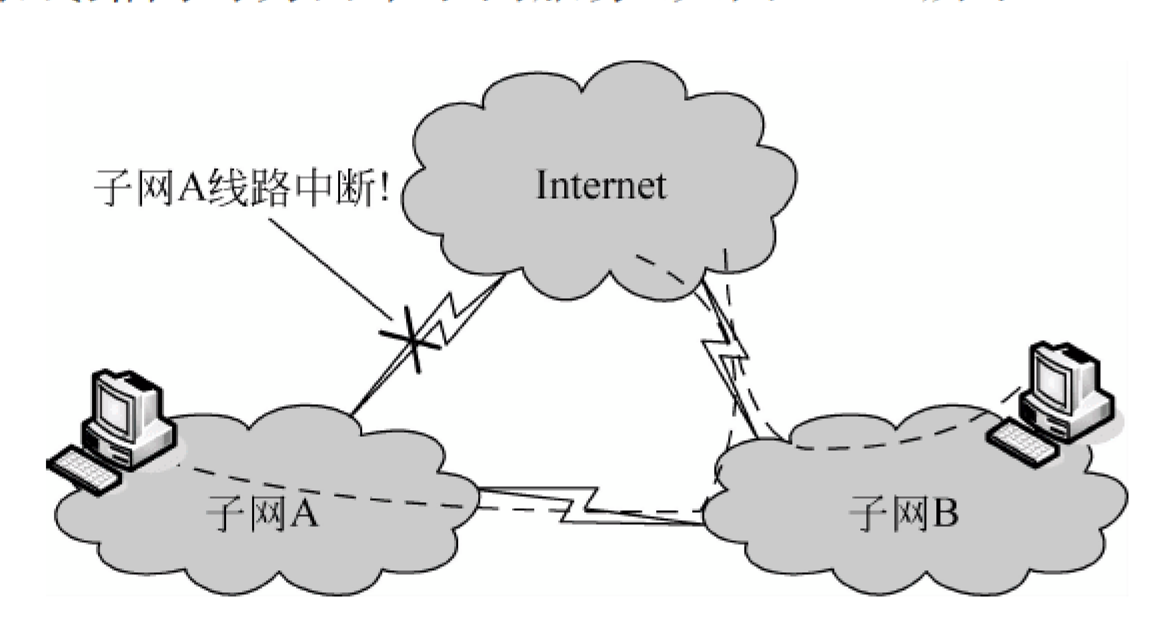


图 17.1　跨网互备线路策略示意

主、备线路的切换有自动切换、人工切换两种方式。自动切换由设备监视和判断故障状态，自动启用备份线路；人工切换则依赖操作员更换线路（或接口）。热备方式一般以自动切换为主，冷备方式以人工切换为主。

线路冗余应区别内部网络的线路和外部网络的线路。在内部网络（通常是局域网）中，线路保护比较容易，线路冗余以规模扩展、性能提升为主要目的，当然在一些重要的信息系统中，也需要冗余线路用以更好地保障系统的可靠性；在接入 Internet 线路或在城域范围构造网络方面，则有不同的线路备份要求。我们主要关注 Intranet 构建和 Internet 接入，讨

论以下几种典型的线路备份方案。

1. 拨号网络备份

如果网络使用专线(或称租用线)方式接入 Internet 或进行网络互连,如 FR 专线、光纤等,则可以采用低速拨号线路作为备份方式。

(1) PSTN 电话网 Modem 拨号上网(需 ISP 提供服务),带宽≤56Kb/s。

(2) ISDN 拨号上网或专线互连,带宽为 64Kb/s 或 128Kb/s。

办公室或家庭通过 ADSL、CableModem 等方式接入 Internet,也可以采用拨号网络作为备份通道,应对网络服务中断情况。

2. 跨运营商备份

冗余备份线路的选择应以相关性较低为原则。

通常认为,同一运营商(或 ISP)提供的多种接入方式具有的关联性太强,即共性的成分太高,不宜作为相互的备份,而应选择独立性强的不同服务提供商(如图 17.2 所示)。

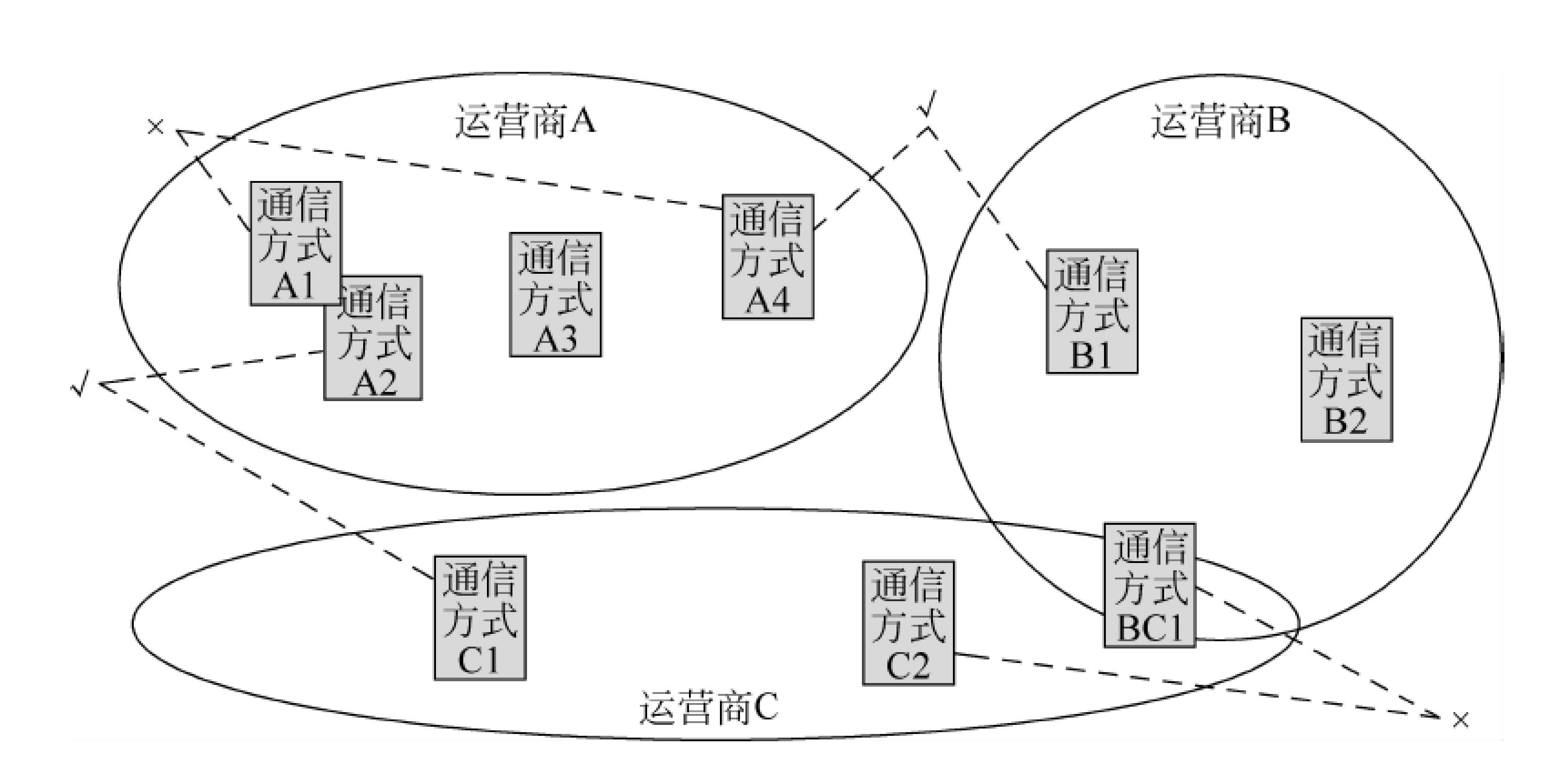


图 17.2 跨运营商备份线路示意

3. 无线通信备份

使用无线通信方式作为备份是个很好的选择,对有线通信网络是一种良好的补充,避免发生线路物理中断的故障。然而目前商用的无线通信接入服务相对较少,而且比较昂贵,这对于备份线路而言是不利的。

可选的无线接入方式包括 Satellite、WLAN/WiFi、WiMax、GPRS/CDMA1x、CDPD 等。发展中的 3G 或 4G 网络也可以成为较好的无线接入方式的选择。

4. 环路技术备份

环路构造技术虽然不一定增加双份的线路,却能带来很好的冗余备份效果,因此是一种较好的技术方案。如图 17.3 和图 17.4 所示,环路构造技术分单环和双环两种方法,都能够有效地提高网络系统的容灾能力。当通信线路在任何一点发生故障,都不会影响网络系统的畅通。但环路技术的运用需要受网络技术和网络设备是否支持的限制,所以应用场合受到一定局限。

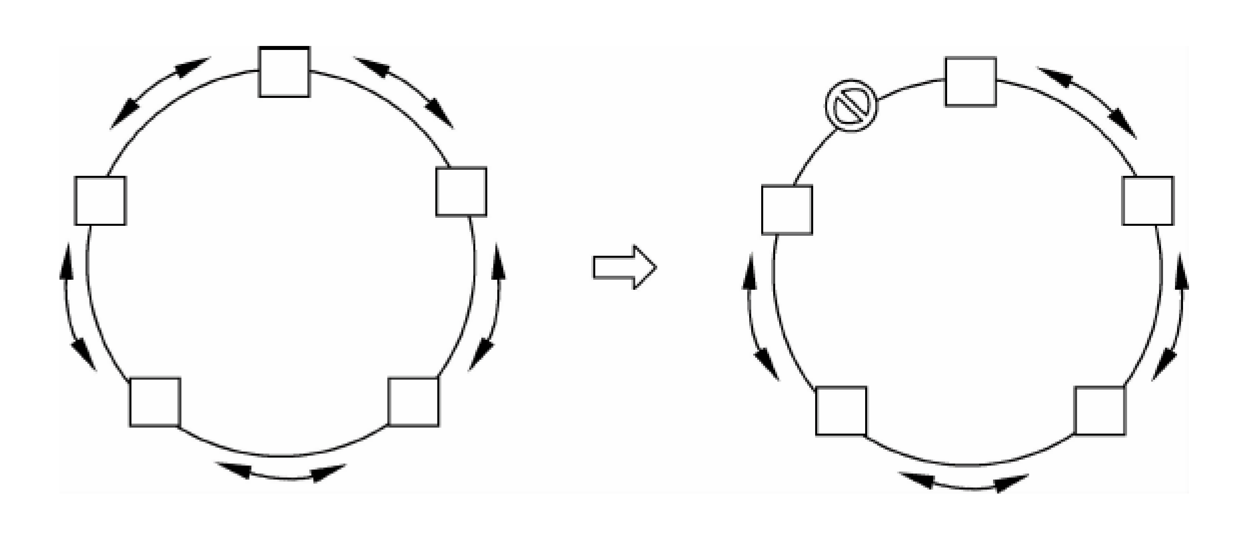


图 17.3　单环容灾原理

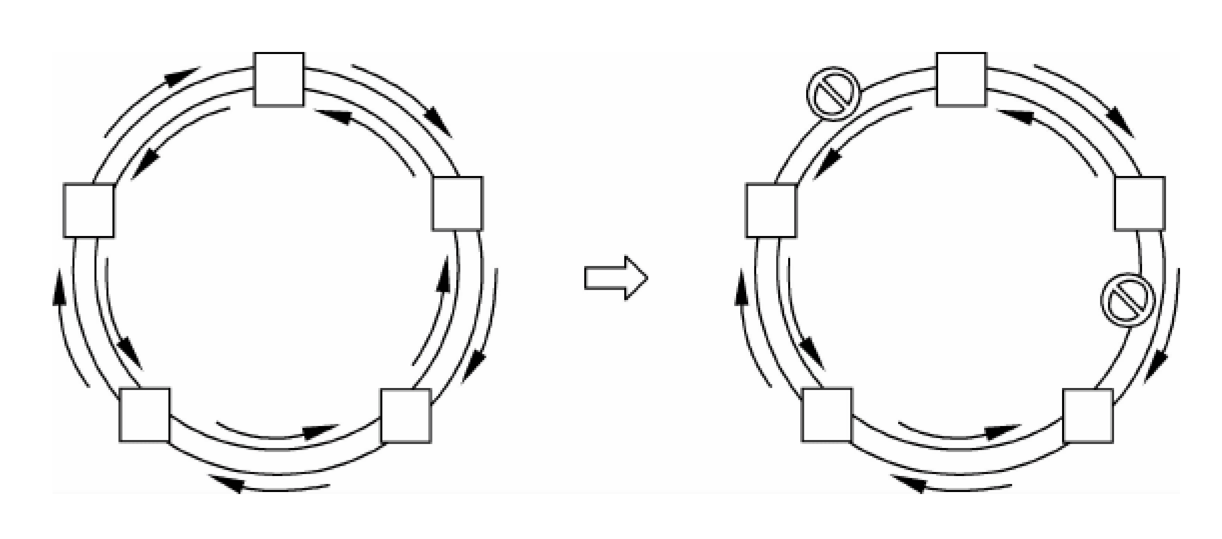


图 17.4　双环容灾原理

17.2.2　路由冗余

路由冗余在一定程度上和线路冗余相同，都是为数据通信提供备份机制，保持在故障情况下的通信能力。但路由冗余是虚拟的(逻辑的)路径构造，它建立在线路冗余的基础上，没有线路冗余，就不可能有路由冗余，因此可以沿用线路冗余的思想和技术，而路由冗余也具有自身所特有的技术思路。

首先，路由冗余应当防止循环路由的情况发生(如图 17.5 所示)。一般通过路由协议的生成树算法来避免路由死循环和路由振荡。

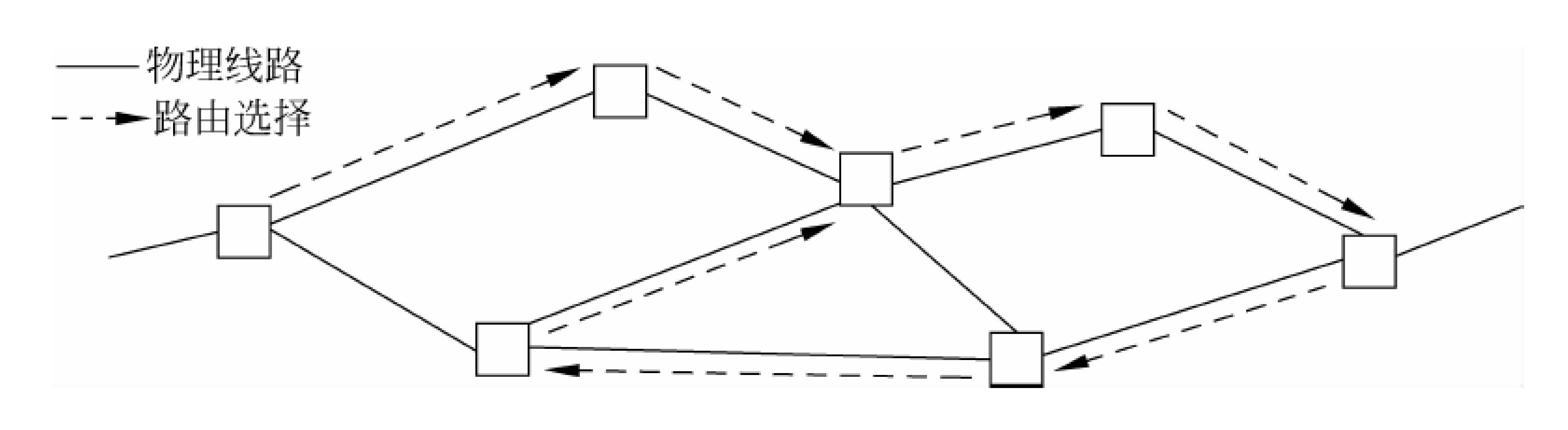


图 17.5　循环路由示意

其次，如图 17.6 所示，应分析并避免假性冗余路由，即在故障情况下无法承担备份使命的路由构造。不过，从线路和设备可靠性角度来分析，可以允许“部分冗余”的冗余路由，即只对容易发生故障的网络区域规划的冗余路由。

构造冗余路由的基础是在物理连接上建立**生成树**(spanning tree)。如图 17.7 所示，生成树算法旨在充分利用多个路由(转发)选择，创立一个有方向性的、无回路的逻辑关系。数据报文即可以通过生成树形成的脉络进行有序的流动。在生成树中，每个结点就是路由器(网关)。在生成树中，每个结点拥有以自身为根结点(root)的子树，是整个生成树的一部分。

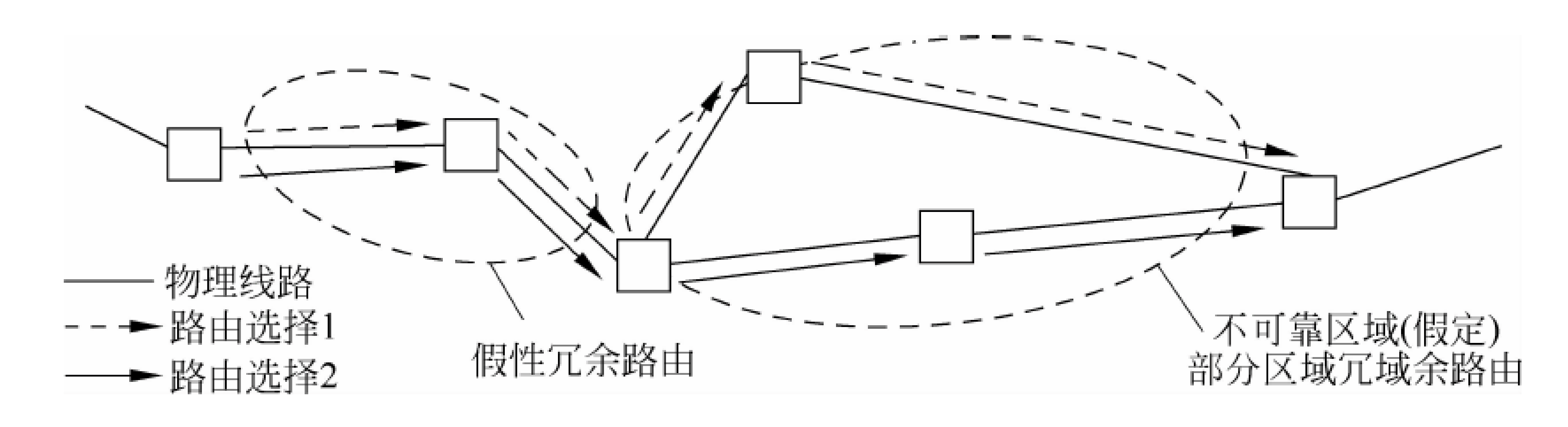


图 17.6　假性冗余路由和部分冗余路由原理

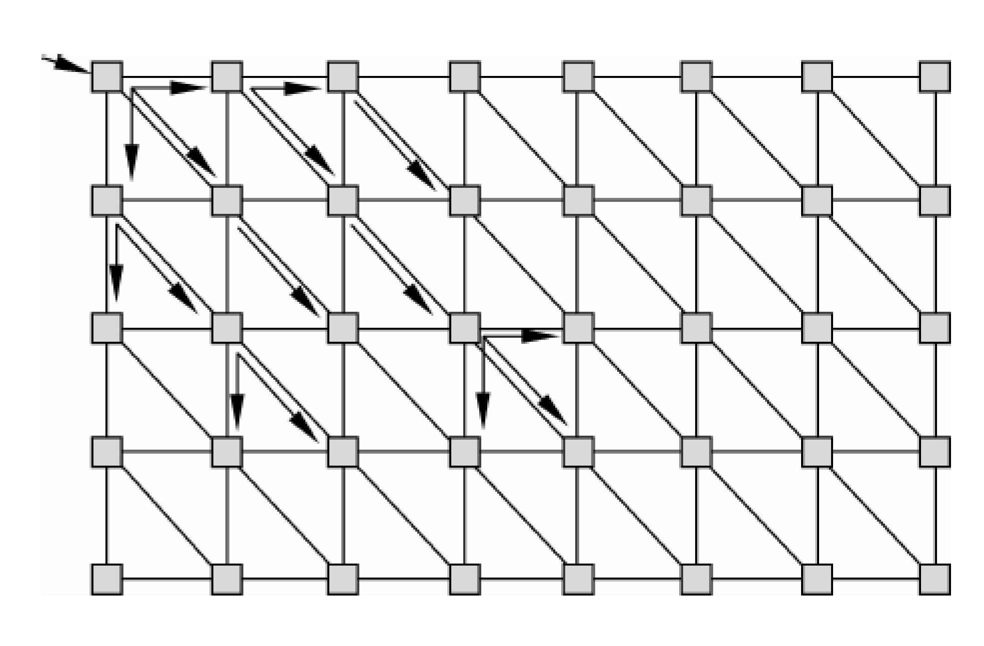


图 17.7　生成树示意

17.3　设施冗余

在一些对可靠性比较敏感的网络系统设计中,部署冗余设备是行之有效的方法,如冗余的服务器、冗余的交换机等。为提高设备(通常指比较昂贵的设备)的利用率,冗余设备同时也起到负载均衡的作用。冗余设备间的控制关系及采用的相关技术,与线路冗余技术有一定相似之处。

为及时相互了解其他设备的工作状态,冗余设备往往通过特殊的**心跳线**(heart-beat line)互连,形成物理的(带外的)或逻辑的(带内的)故障侦测通道。如图 17.8 所示,系统运行中,每个设备中的特殊进程或应用程序嵌入进程不断向其他兄弟设备发送心跳信号,当然也在不断接收、侦听对方的心跳信号,一旦发现由故障引起的“心跳停止”,则可立即接管对方设备的工作。

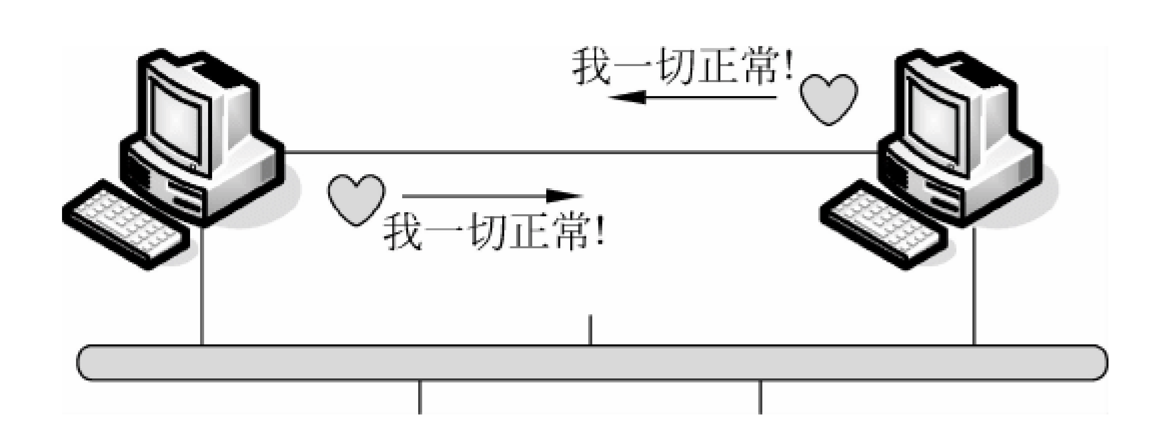


图 17.8　冗余设备心跳线示意

在 Internet 网站的外部网络和内部网络的构造中,设备冗余技术的运用比较普遍。

一种常用的部署方案是服务器集群。采用大量的 Web 服务器、E-mail 服务器向用户提供服务,既可以同时服务于大批用户,又可以相互备份,任意一台服务器故障都不会明显影响网络访问。

另一种方案通常适用于一些后台核心系统，为双机冗余系统(如图 17.9 所示)。需要注意的是，通常需要对通信环节上的每一个设备进行冗余部署，才能真正达到消除单点故障的目的。

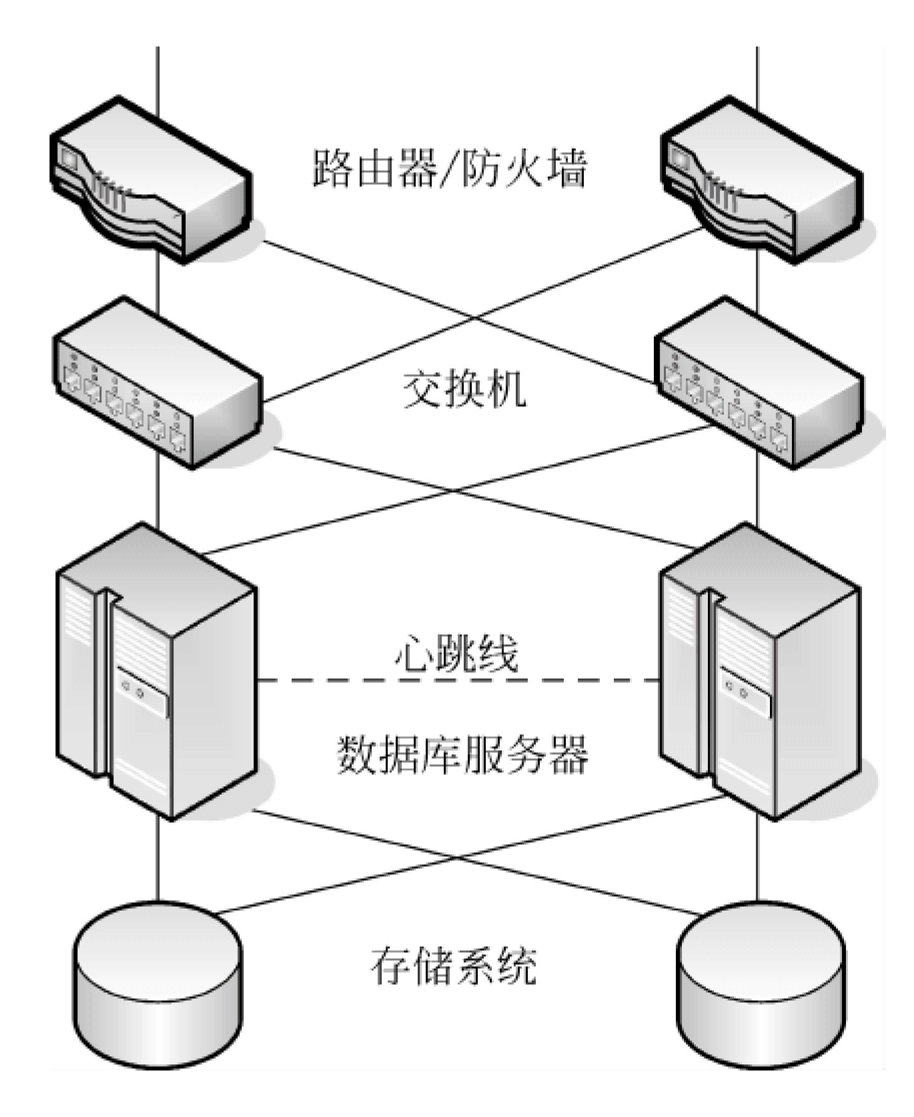


图 17.9　双机热备系统示意

17.4　存储冗余

17.4.1　RAID

独立磁盘冗余阵列(Redundant Array of Independent Disks，RAID)既是一种存储介质的组织技术，又是数据可靠存储的技术。RAID 最初的研制目的是组合容量小的廉价磁盘来代替容量大的昂贵磁盘，以显著降低大批量数据存储的费用，同时也希望采用校验和冗余信息的方式，牺牲部分存储空间，但可使得部分磁盘失效时不会使数据受损，即提供一定程度的数据保护，并且适当提升数据传输速度和数据访问效率。

RAID 通过专门的控制器执行存储数据交互协议、控制数据的物理介质存储操作、管理存储系统。RAID 的存储介质组织为一个磁盘阵列，纵向称为块(磁盘组)，横向称为条带(Stripe)。RAID 可以达到每个机柜 512 个磁盘，10TB 的容量，并提供多种技术规范(级别)，如 RAID0～RAID7，一些规范还可以进行组合，如 RAID10、RAID50、RAID53，适应不同存储需求。RAID 各技术指标明显优于 JBOD。

1. RAID0

RAID0(条带化，striping)将数据分成一定的大小的数据块，以条带方式依次存储到不同的磁盘组中(如图 17.10 所示)。

RAID0 可以执行并行读写操作，数据吞吐率大大提高，各个磁盘驱动器的负载也比较平衡。RAID0 不需要计算校验码，易于实现，磁盘空间的存储效率最大(100%)。

但 RAID0 技术的缺点也很明显：因为不提供数据校验和冗余保护，一旦数据损坏，将无法恢复；一个数据块损坏，会造成整批数据失效；最严重的情况是，由于数据都是分布在

不同磁盘组上,一旦一个磁盘组损坏,将可能导致 RAID0 阵列上所有数据全部丢失。因此 RAID0 适用于对读写性能和磁盘空间利用率要求较高,但数据重要性较低的应用。

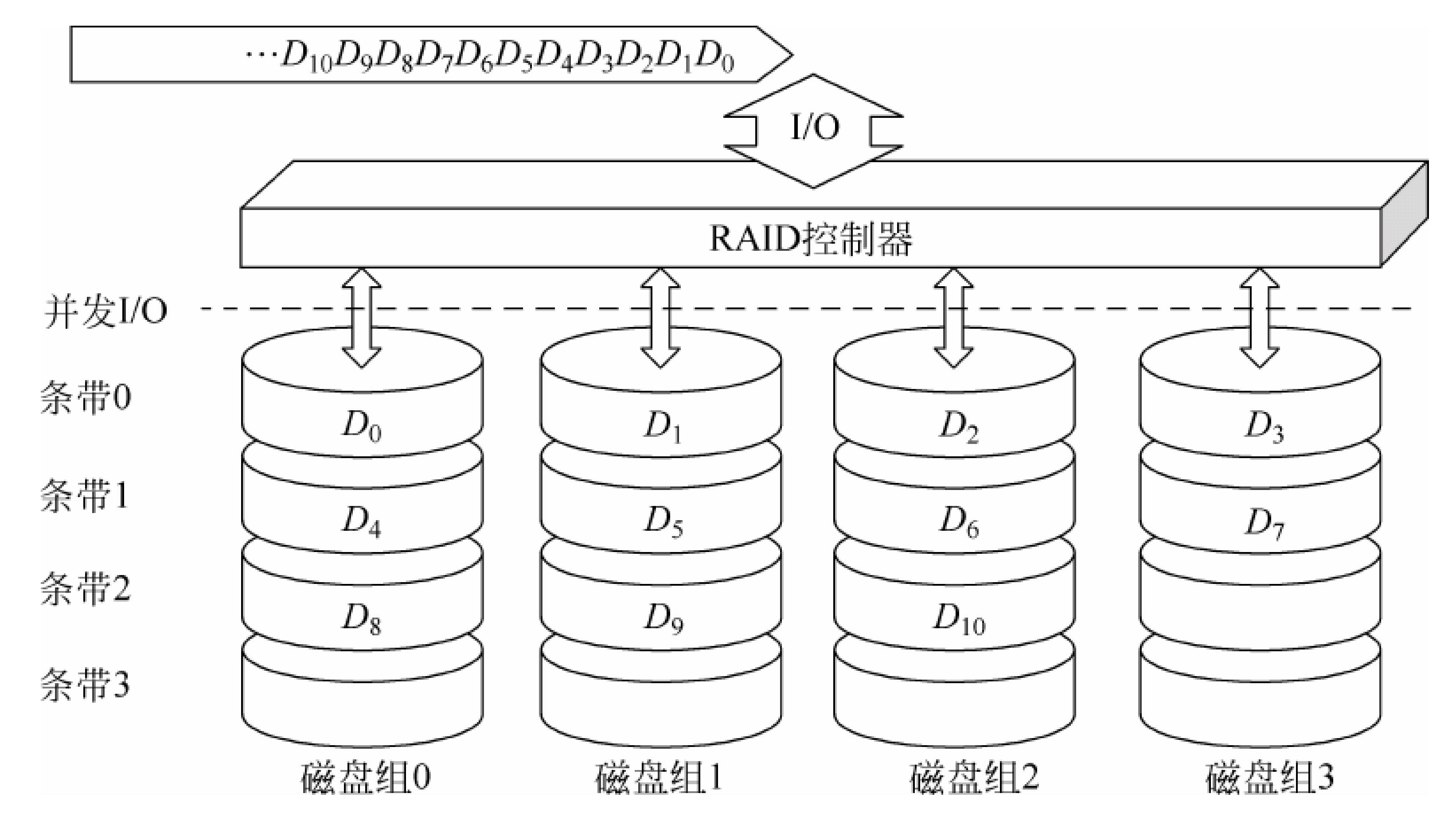


图 17.10 RAID0 原理

2. RAID1

RAID1(镜像,mirror)将数据分别复制到工作磁盘和镜像磁盘(如图 17.11 所示),形成冗余数据,因此磁盘空间利用率为 50%。RAID1 对数据写入时间会有轻微影响,但是对读取操作没有任何影响。

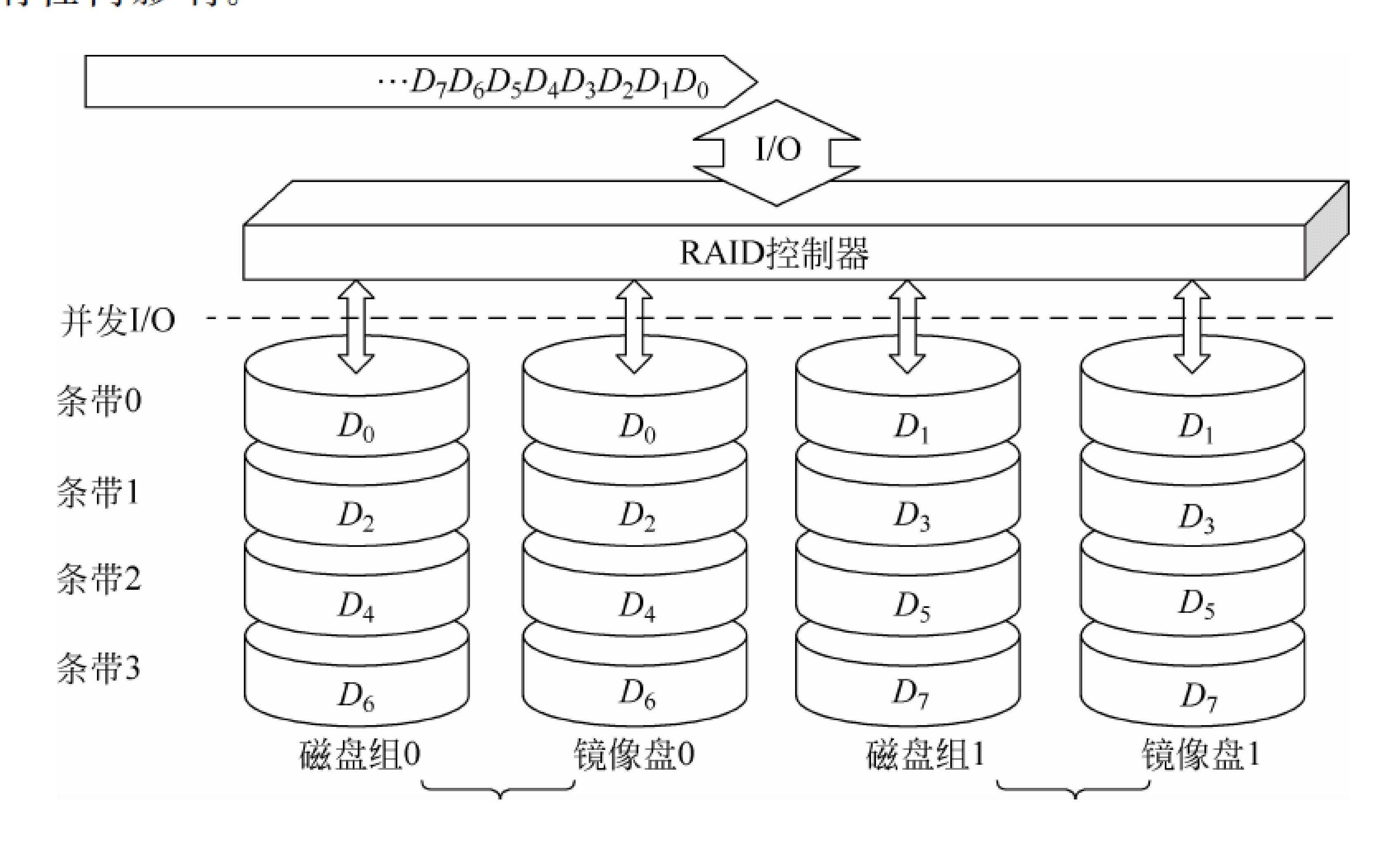


图 17.11 RAID1 原理

一旦有工作磁盘发生故障,RAID1 自动从镜像磁盘读取数据,提高了数据可靠性,适合对数据保护要求较高的应用。两组以上的 RAID1 并发操作又称为 RAID0+1。

RAID1 技术还支持热替换(hot swap)操作,在不断电的运行情况下对故障磁盘进行更换,更换完毕只需从镜像盘上恢复数据即可。

3. RAID2

RAID2 为**纠错海明码**(Hamming code)磁盘阵列,主体思想与 RAID3 类似,将数据条

带化分布于不同的硬盘上，数据块单位为比特或字节。阵列中序号为 2^i，$i=0,1,\cdots$ 的磁盘（第 1，2，4，8，…）作为校验盘，存放校验数据，其余的磁盘用于存放数据，磁盘数目越多，校验盘所占比率越少。

由于海明码为一种错误校正码（Error Correction Code，ECC），因此 RAID2 可以在数据发生错误的情况下自动修正错误，以保证正确的输出。

RAID2 在大数据量存储情况下性能很高，但技术实施较为复杂，因此实际应用较少，已被 RAID3、RAID4、RAID5 取代。

4. RAID3

RAID3（带奇偶校验码的并行传送）采用一个硬盘组作为校验盘，其余磁盘组作为数据盘，数据以 b 或 B 为单位，按条带依次存放到各个数据盘中。不同磁盘组上同一条带的数据做奇偶校验，并把校验值存放在校验盘中（如图 17.12 所示）。

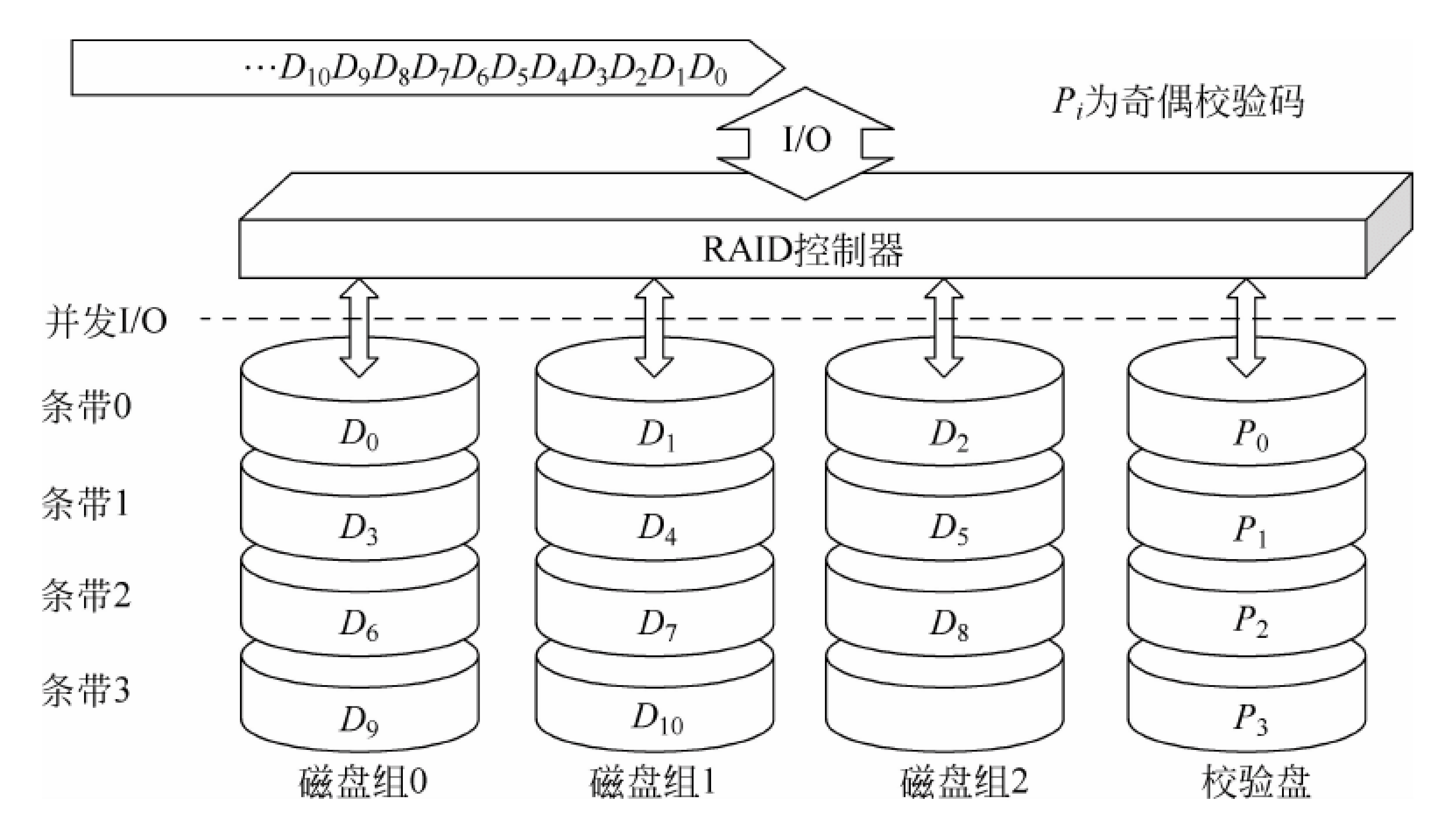


图 17.12　RAID3 原理

当 RAID3 中有一个数据盘出现损坏，可以读取同一条带的其他数据块，然后根据校验值恢复受损数据。

RAID3 采用并发读写方式，读操作性能与 RAID0 一致，而且提供了数据容错能力，但是，在写操作时性能有所下降，因为每一次写操作，即使是牵扯到某个数据盘上的一个数据块，也会影响同一带区的其他数据，需要重新计算校验值并写入到校验盘中，增加了系统开销。

由于 RAID3 的校验盘在系统进行大量写操作时容易成为性能瓶颈，因此比较适用于侧重大量读取操作的应用，例如 Web 系统、信息查询系统、流媒体系统等。

5. RAID4

RAID4（带奇偶校验码的独立磁盘结构）和 RAID3 类似，不同的是，RAID4 对数据的访问是纵向按磁盘组进行的，无须像 RAID3 那样，每一次 I/O 操作都要涉及所有磁盘组。但 RAID4 在差错恢复时难度比 RAID3 大，控制器的设计难度也较大，访问数据的效率不够高。

6. RAID5

RAID5 为浮动式校验盘的条带化存储（striping with floating parity drive）技术。数据

以 b 或 B 为单位,以条带化形式存放,可以进行并发读写操作,对写入数据进行校验,在不同的磁盘上存储校验码(如图 17.13 所示)。当有一个磁盘损坏,可以利用条带上的其他数据和校验码进行恢复。

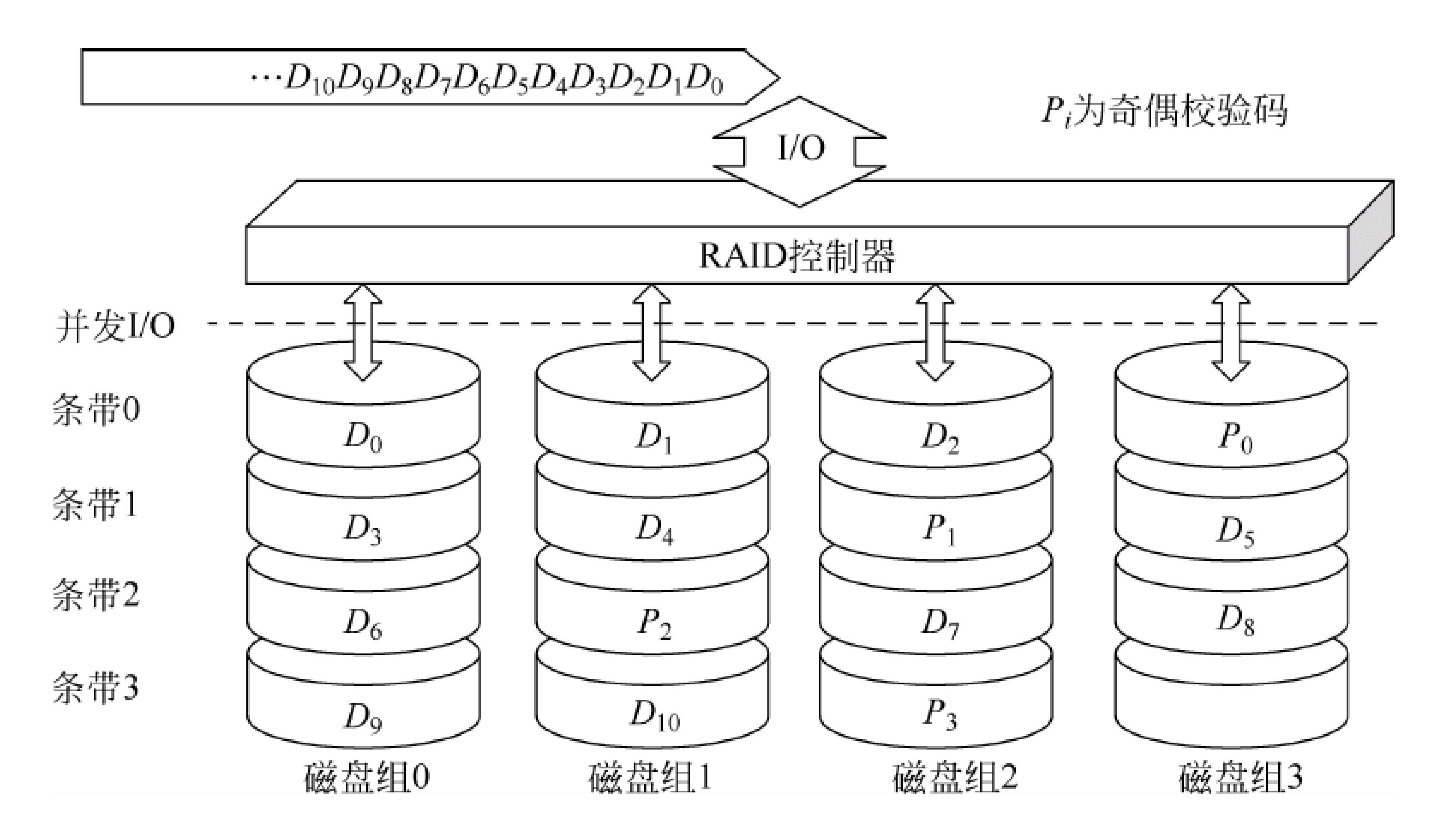


图 17.13 RAID5 原理

由于 RAID5 没有固定的校验盘,而是把奇偶校验信息均匀地分布在阵列所属的各个硬盘上,因此在每块硬盘上,既有数据信息也有校验信息。这一改变解决了争用校验盘的问题,有利于并发操作,提高访问效率。所以 RAID5 既适用于大数据量的操作,也适用于各种事务处理,是一种效率、容量、成本和容错性能兼顾的磁盘阵列。RAID5 技术因其良好的技术特性,得到广泛的应用。

7. RAID6

RAID6(带有两种分布存储的奇偶校验码的独立磁盘结构)是对 RAID5 的扩展,增加了第二个采用不同算法的校验码,即使两块硬盘同时失效,也可以进行数据恢复,极大提高了数据可靠性。然而,RAID6 较差的性能和复杂的实施方式影响其投入实际应用。

8. RAID7

RAID7(优化的高速数据传送磁盘结构)采用独立的操作系统进行存储访问管理,可看做是一台存储计算机(Storage Computer),与其他 RAID 技术有明显区别。

17.4.2 SAN

存储区域网(Storage Area Network,SAN)是通过专用网络或线路(通常是高速光纤网络)连接存储设备和服务器(群)的存储系统部署方式(如图 17.14 所示)。当有数据的存取需求时,可以经由 SAN,在联网服务器(群)和存储设备之间高速传输。

SAN 包括以下三大类组件:服务器 SAN 接口和驱动、SAN 和 SAN 存储设备。

(1) 服务器通过网卡接入 SAN,实现网卡的驱动以及 SAN 协议,为应用程序或操作系统提供数据存储读写接口。服务器群组可以通过 SAN 共享存储资源,数据读写操作并不占用承载网络带宽,也不占用服务器计算资源。

(2) SAN 为光纤网络,采用光纤集线器、光纤交换机等设备组网,保证了数据传输的可

靠性和数据交换效率。多数 SAN 都基于 FC 体系结构，而 SCSI 在 FC 中以一种高层协议的方式出现，使服务器可以用比较熟悉的技术访问存储器。

(3) SAN 存储设备则提供海量的存储空间、存储管理以及 SAN 接口，便于进行管理、维护和扩展。存储介质的组织可以采用 RAID 技术。

SAN 得以大量应用，得益于其性能、存储管理和可扩展性上存在的优势。典型的 SAN 应用场合包括以下几种。

(1) 数据存取——由于存储设备的中心化，大量的文件服务器、数据库服务器可以低成本存取和共享信息，同时系统性能不会明显下降。

(2) 存储共享——两个或多个服务器可以共享一个存储单元，这个存储单元在物理上还可以分成多个部分，而每个部分又连接在特定的服务器上。

(3) 数据备份——通常的数据备份都要依赖于承载应用的局域网或广域网设备。通过使用 SAN，这些操作可以独立于原来的网络，从而能够提高备份操作的性能。

(4) 灾难恢复——当灾难发生时，使用 SAN(而不是传统的磁带)，可以采用更多手段实现数据的自动备份。而且这种备份可以是热备份形式，一旦数据出错，立即可以获得该数据的镜像内容。

虚拟 SAN(Virtual SAN，VSAN)技术可以在一个单一的 SAN 结构中创建多个仿真硬件的独立环境，每个 VSAN 都可以作为一个常规的 SAN 进行单独分区，拥有独立的数据交换服务，相互隔离，从而提高 SAN 的灵活性、安全性、可扩展性和灾难恢复能力。

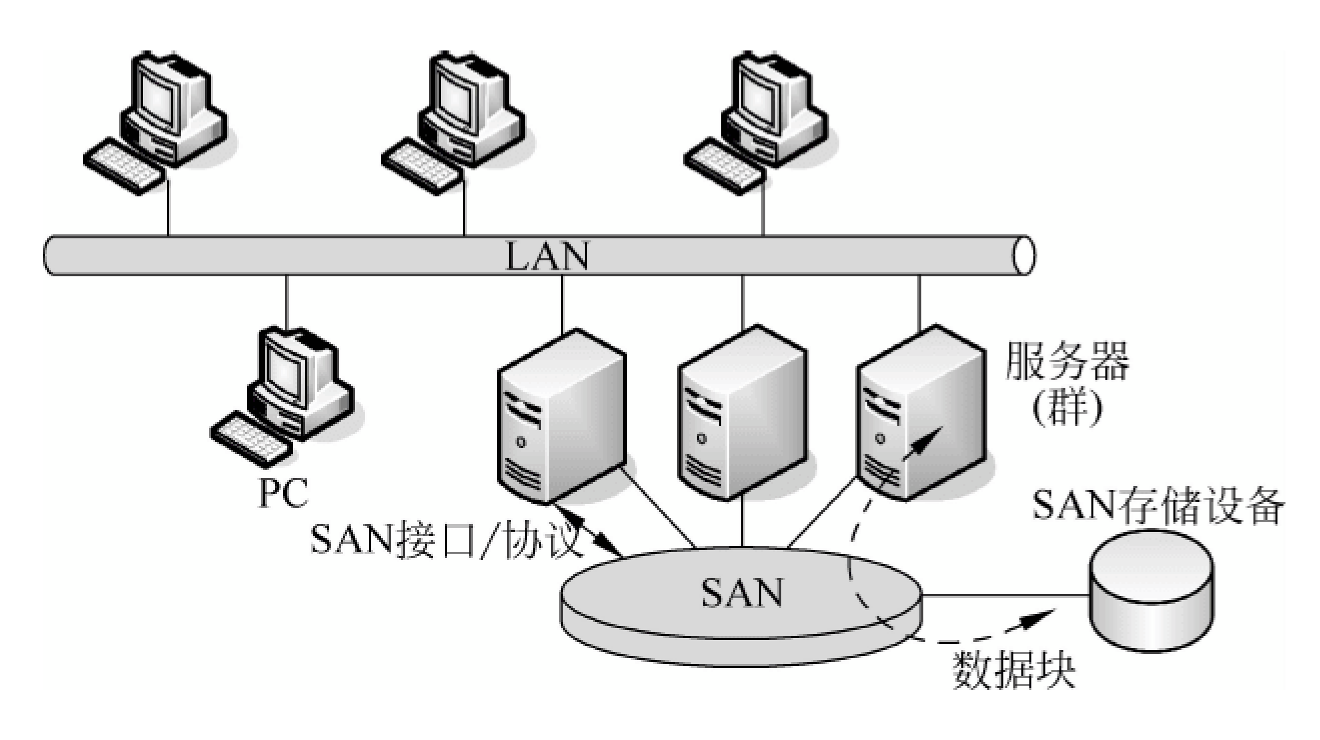


图 17.14　SAN 系统结构示意

17.4.3　NAS

网络附着式存储(Network Attached Storage，NAS)是一种专用的数据存储设备，因此也称为网络存储器。

NAS 以数据为中心，将存储设备与服务器彻底分离，集中管理数据，从而提高整体性能，降低总拥有成本，有利于数据管理。与 SAN 相比，NAS 服务器不需要安装专门的数据存储接口，也不需要部署专用的光纤存储网络，安装和使用十分便捷。

NAS 系统(如图 17.15 所示)包括存储介质(例如磁盘阵列)、控制主机和内嵌系统软件，直接连到业务承载网络上，不再挂接于服务器后端，增加和移除 NAS 服务器操作不会中断网络的运行。

对照 SAN 面向数据块的存取操作，NAS 更加适合文件级别上的数据处理，提供跨平台

文件共享，将相关处理负荷从应用或企业服务器上卸载下来，但进行数据读写操作需要占用业务网络带宽资源。

NAS存储系统相当于一个C/S系统架构下的服务器，拥有独立的处理器、操作系统或微内核，通过TCP/IP网络交换控制信息和数据，运行UNIX系统的网络文件系统(Network File System，NFS)或Windows系统的公用互联网文件系统(Common Internet File System，CIFS)等文件I/O协议，管理存储系统与其客户机之间的数据传输。

NAS和SAN可相互配合，在两个方面提供应用功能的互补。

(1) NAS产品可以在特定的SAN中为文件传输提供优化的性能。

(2) SAN可以扩展为包括TCP/IP在内的非存储关联的网络协议。

事实上，尽管SAN和NAS存在技术、操作上的差异，但分界线正在变得越来越模糊。

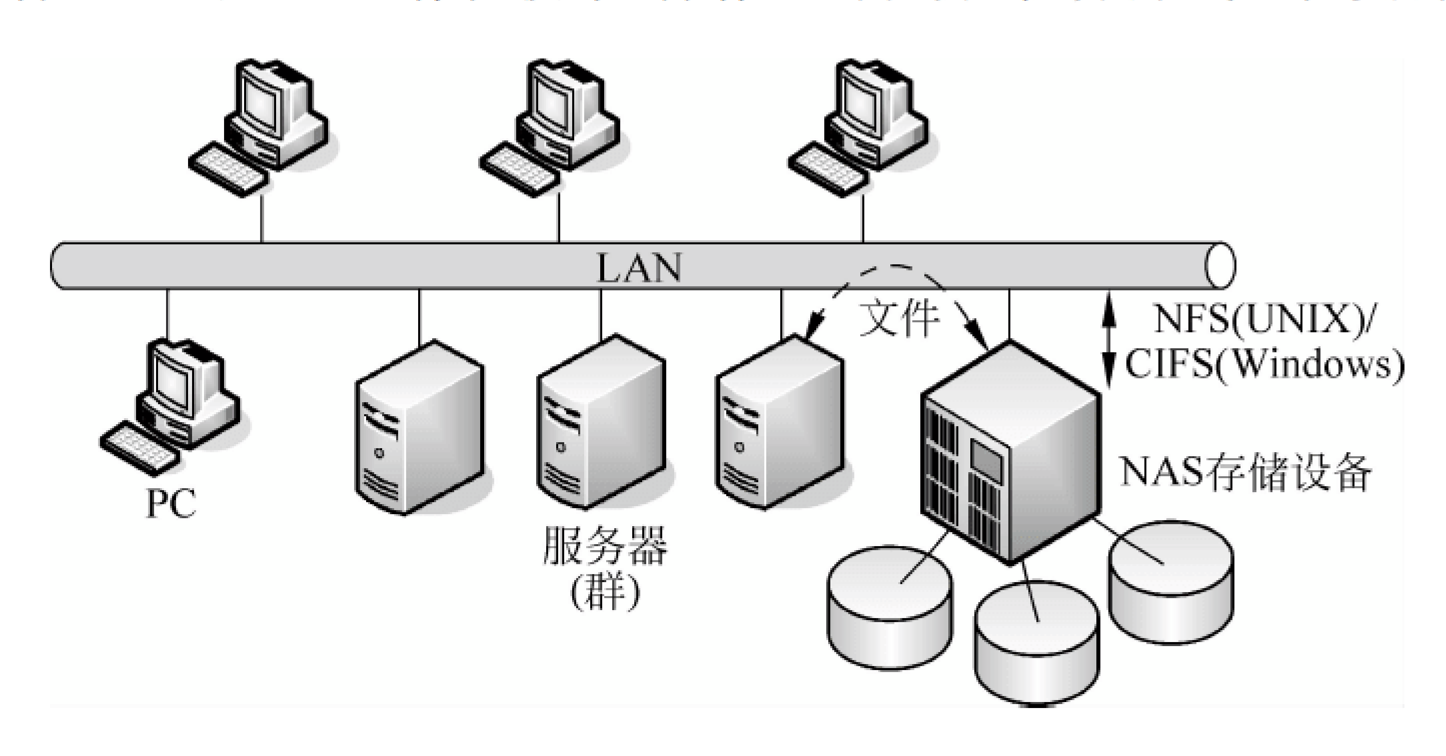


图17.15 NAS系统结构示意

17.4.4 SoIP

基于IP的存储(Storage over IP，SoIP)是存储技术和存储网络发展的技术方向之一。IP已经被SAN和NAS采用，网络应用的全球化趋势也迫使存储服务能够走出局域网覆盖的局部范围，扩展到Internet上。

以海量信息内容为荣的Internet历来对存储业务非常渴求，不论是个人、企业或服务运营商，都对Internet存储趋之若鹜。主要体现为两种类型的需求：一是显性存储，对存储空间大小比较直观和敏感，有特定的目的，如网络硬盘、个人相册、文件共享、电子邮件、信息保全等；二是隐性存储，存储需要通过网络应用系统间接地反映出来，应用目标的通用性较强，如银行账务、博客空间、媒体播放、软件租用等。针对不同的Internet应用，网络存储技术提出了不同的解决方案，使这一领域十分活跃，相关的存储技术一直在改进、完善和创新中。

互联网分布式存储(Internet distributed storage)并非指采用分布式数据管理方法的存储系统，而是指在Internet上特有的P2P(对等网络)存储模式。特别是海量的视频、音频、动画、图片等多媒体文件，分散保存在网络用户的计算机上，通过应用系统进行内容聚合，按需提供高效的、就近的下载服务。

虚拟存储(virtual storage)是整合Internet上各种物理存储系统，形成一个整体，以标准化(或一致性)的方式，提供永久保存数据服务，或提供能被用户调用的功能，使存储服务

成为 Internet 重要的公共基础设施。

使用虚拟存储服务的用户在获得数据保存的同时，并不需要知道自己的数据被保存在哪里，是以何种方式被存储的。虚拟存储大致分为 3 种形式：基于主机或服务器的虚拟存储、基于存储系统的虚拟化和基于网络的虚拟化。不论哪种形式，虚拟存储技术都需要考虑存储的可靠性、访问的高效性、操作的便利性、容量的扩展性、功能的丰富性、信息的安全性、计算的移动性、平台的无关性、管理的有效性、系统的容错性等方面因素。

17.5　数据冗余

随着计算机存储代价（每比特投入）的不断降低，同时系统数据的价值越来越宝贵，数据的冗余既十分必要，也有很大的可行性。

数据冗余，往往也被称为数据备份，不仅是把数据“复制一份保存”那么简单。如果保存的数据是不完整的、难以恢复的，冗余就失去了意义。所以，数据冗余是一个系统工程，包含了从系统设计、数据规划、数据备份、数据恢复到数据管理等各个方面和环节。其次，数据备份工作应该和系统性能优化结合起来考虑，在提高数据抗毁性（容灾能力）的同时，也提高了系统的操作性能。

（1）数据冷备：利用磁盘（磁盘阵列）、磁带机、光盘等存储设备，实时地或定时地备份系统数据，在需要时可反向恢复，最大限度地减小由于“天灾人祸”造成的数据损毁。

（2）数据热备：系统同时保存和维护两套数据副本，相互严格同步。正常情况下以其中一个数据库为主进行操作（主要是读操作），而当一个出现故障时，可立即切换到另一个数据库进行操作，并尽快修复故障数据库。

（3）异地备份。异地备份是数据容灾工作最重要也是最有效的手段（如图 17.16 所示）。从理论上说，两地相隔越远，关联性越小，容灾效果越佳。异地备份可以选择数据冷备或热备的方式，但不管是哪一种类型，都需要有比较高的系统投入，尤其是通信线路条件的保障，应根据系统需求进行合理规划。

数据备份可以利用数据库系统提供的功能，也可以自行设计应用系统的控制逻辑，跨数据库实施。

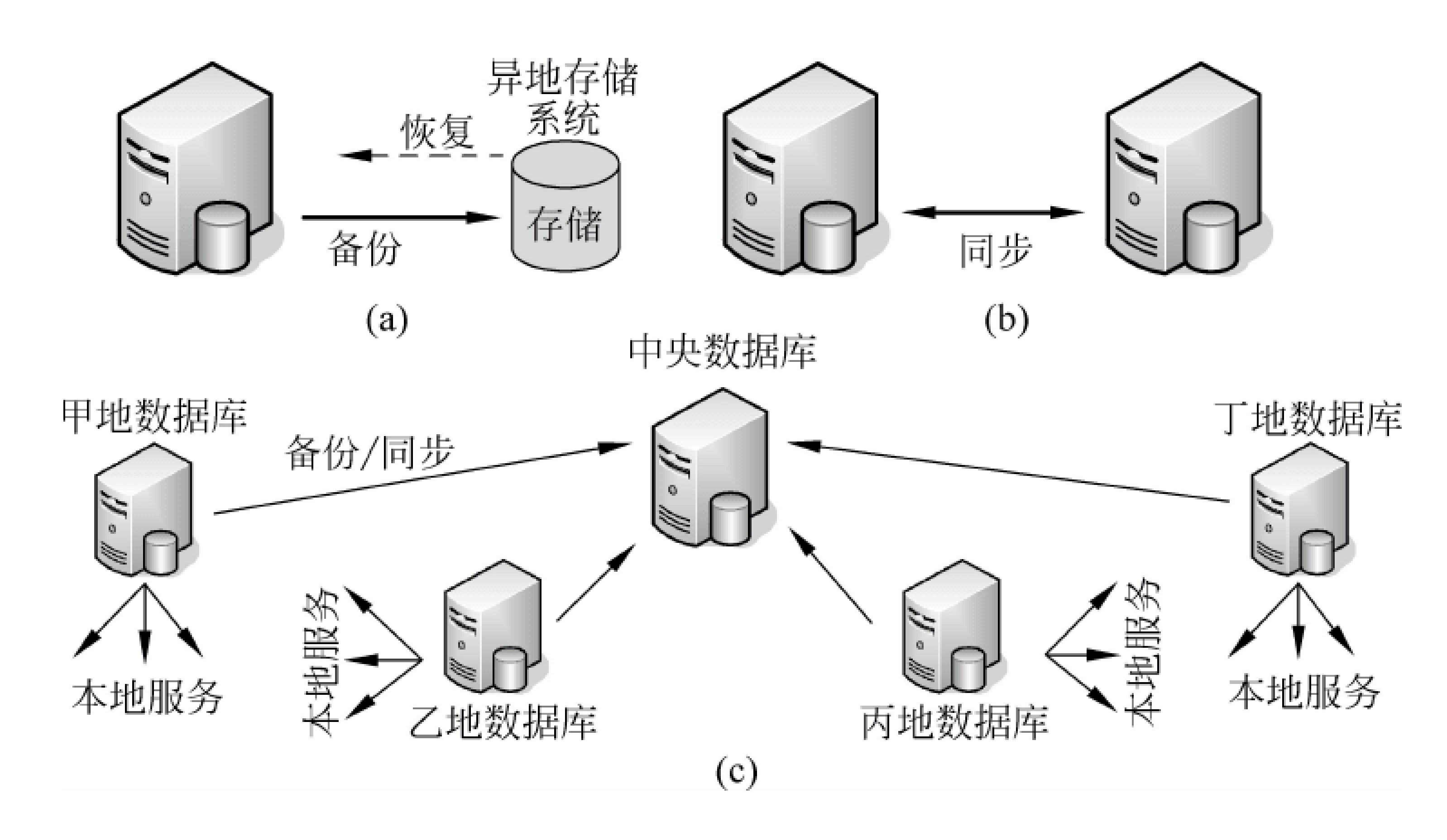


图 17.16　数据库异地备份示意

CHAPTER 18

第 18 章 网络技术发展

18.1 物联网

18.1.1 物联网原理

Internet 是一个把人与人联系在一起的网络。然而，对于人类社会的需要，这似乎显得不够。无线通信、传感器、嵌入式处理机、电子标签、信息安全、协议和软件等技术的发展，使得所有 M2M 的连接成为可能，即进一步实现现实世界中人与物(Man to Machine/Material)、物与物的互连互通，这就是**物联网**(Internet of Things)概念的由来，同时也是物联网的应用目标。

英语里 things 是各种东西的总称，除了智能化的人和计算机，也应该包括所有非智能化的东西，如桌椅、门窗、家电、道路、环境等。对非智能物品的智能化同样是物联网的研究范畴。

物联网技术的提出不仅为计算机网络拓展了新的发展领域，而且为网络应用开辟出巨大的想象空间。智能交通、智能家居、智能汽车、智能服务、智慧社区、智慧城市等目标将是引导网络技术发展的新线索。

物联网的核心技术是电子标签、传感器和无线通信。电子标签用于标识非智能体，使之拥有独特的电子身份证，支持读取和记录信息的功能，非智能体因此具备了智能；传感器采用电子技术感知各类环境信息，如温度湿度、风力风向、开启关闭、空间位置等，实现从模拟信号到数字信号的转换(A/D)或进行逆向控制(D/A)，可供计算机进行数字化、智能化处理；数字无线通信则是各种信息相互传输的共性需求，可以摆脱线缆的束缚，使应用更自由、更灵活、更自然、更人性化。

18.1.2 RFID

射频标签(Radio Frequency Identification，RFID)，常称为非接触卡或电子标签，是一种通过无线通信方式进行信息读取和记录的小型器件。

与一维或二维条码、IC 卡、生物特征(指纹、虹膜)等识别技术相比,RFID 具有免接触数据传输的特点,避免了接口易污损的缺陷,容易操作,且不需要自带电源,免维护,工作性能稳定,可靠性高,可以根据需要封装为各种形状,成为一种用于标识物品的有效手段。随着 RFID 成本的不断降低,其应用面将逐步扩大,甚至可以对每一样物品进行标识,方便了使用和管理。因此,RFID 成为智能化、流水线、商业贸易、现代物流等物联网领域的重要技术。

无源 RFID(passive RFID)的技术特色在于无电源、无线的工作方式。读写器发送的电磁波能量一部分被 RFID 转化为电能,供给 IC 运行,激活的 IC 即可接收无线信道上的载波数据进行处理,并向读写器发送响应信息。因此,RFID 一般以近距离、低速率通信为主,传输数据量也较少。

根据 RFID 的不同技术和结构、面向不同应用,可分为如图 18.1 所示的多种类型。

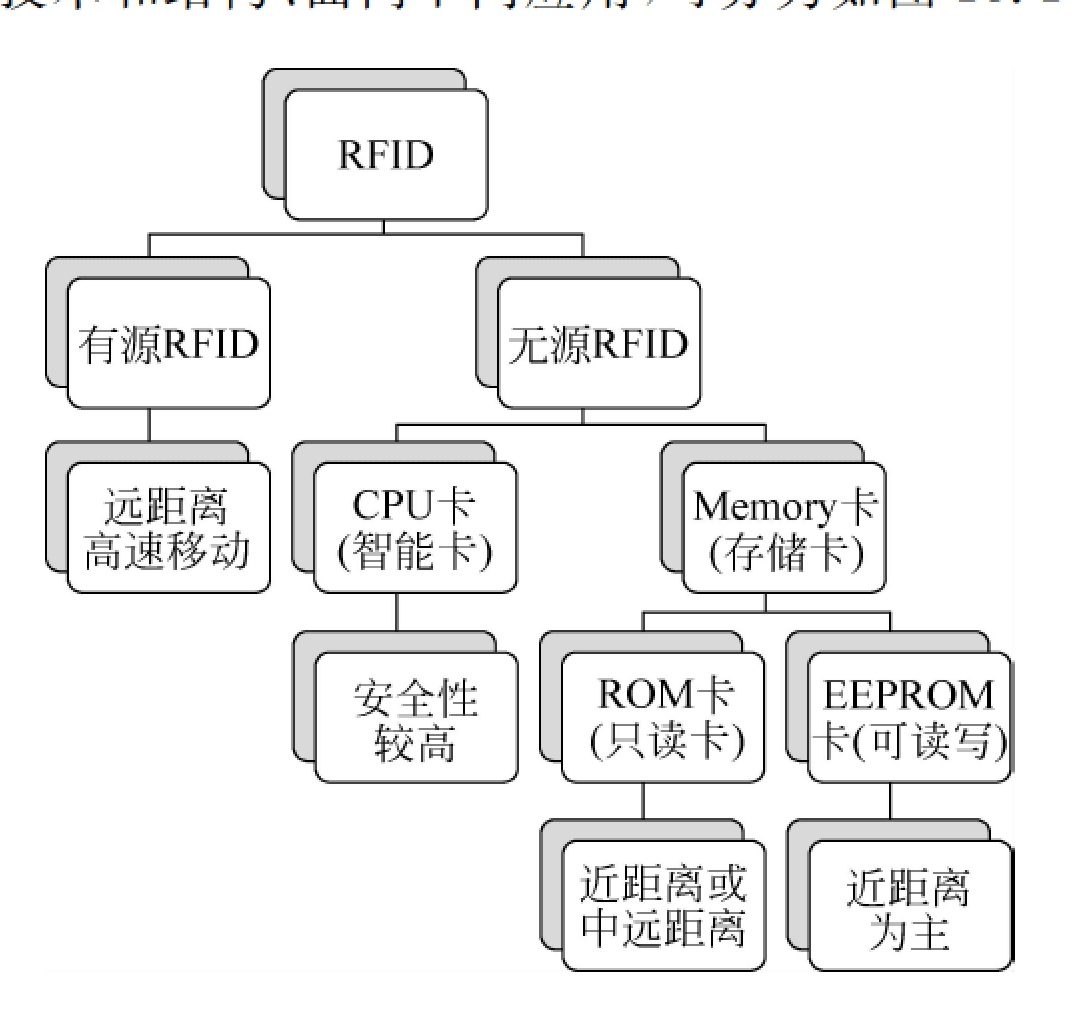


图 18.1　RFID 分类

近距离 RFID 卡最为常用,工作频率小于 30MHz(如 125kHz、13.56MHz),具体可分为密耦合卡(Close-coupled IC Card,CICC),作用距离小于 1cm;近耦合卡(Proximity ICC,PICC),作用距离在 7～10cm;疏耦合卡(Vicinity ICC,VICC),作用距离小于 1m。远距离 RFID 系统可达 1～10m 或更远距离,在超高频及微波范围工作,如 915MHz、2.4GHz、24GHz 等。

RFID 技术相关国际标准有 ISO/IEC 10536、14443、15693、18000—6 等以及 EPC global(Electronic Product Code global)厂商联盟制定的规范。

常用的 ISO/IEC 14443 卡有 TYPE A 和 TYPE B 两种类别,在信号调制技术和编码技术等方面有所差异,具体影响到以下几个方面。

(1) TYPE A 的调制信号区别明显,易于检测,抗干扰能力强,但在每一比特的传送(速率为 106Kb/s 时,传送周期为 9.4μs)中,有大约 3μs 的信号间歇,使读写器到卡的能量供应中断,必须在卡内电路中加一个大容量电容以维持一定的能量供应;而 TYPE B(采用 10% ASK)卡片可以从读写器获得持续的能量,但信号区别不明显,容易造成误读/写,抗干扰能力较差。

(2) TYPE A 卡片能量的中断会导致卡片时钟的中断,而回避时钟中断问题又可能成为后门,让单步跟踪等攻击行为有机可乘。

(3) 当试图提速时,若传送速率为 212Kb/s,比特传送周期仅为 4.7μs,这种情况下 3μs 的间歇已大于传送周期的 60%,若传送速率为 424Kb/s,比特传送周期仅为 2.35μs,这种情况下 3μs 的间歇已使系统无法工作,即 TYPE A 无法实现更高的传送速率。

(4) TYPE A 的防冲突需要依赖较精确的时序,因此需要在卡和读写器中分别配置相应的硬件;而 TYPE B 的防冲突可以用软件来实现。

(5) TYPE A 产品拥有较高的市场占有率,且在较为恶劣的工作环境下更有优势;TYPE B 则在安全性、高速率和适应性方面有更好的前景,更适合于 CPU 卡。

以普遍应用的 ISO/IEC 14443 卡(TYPE A)为例,无源 RFID 的工作原理和组成结构见图 18.2。

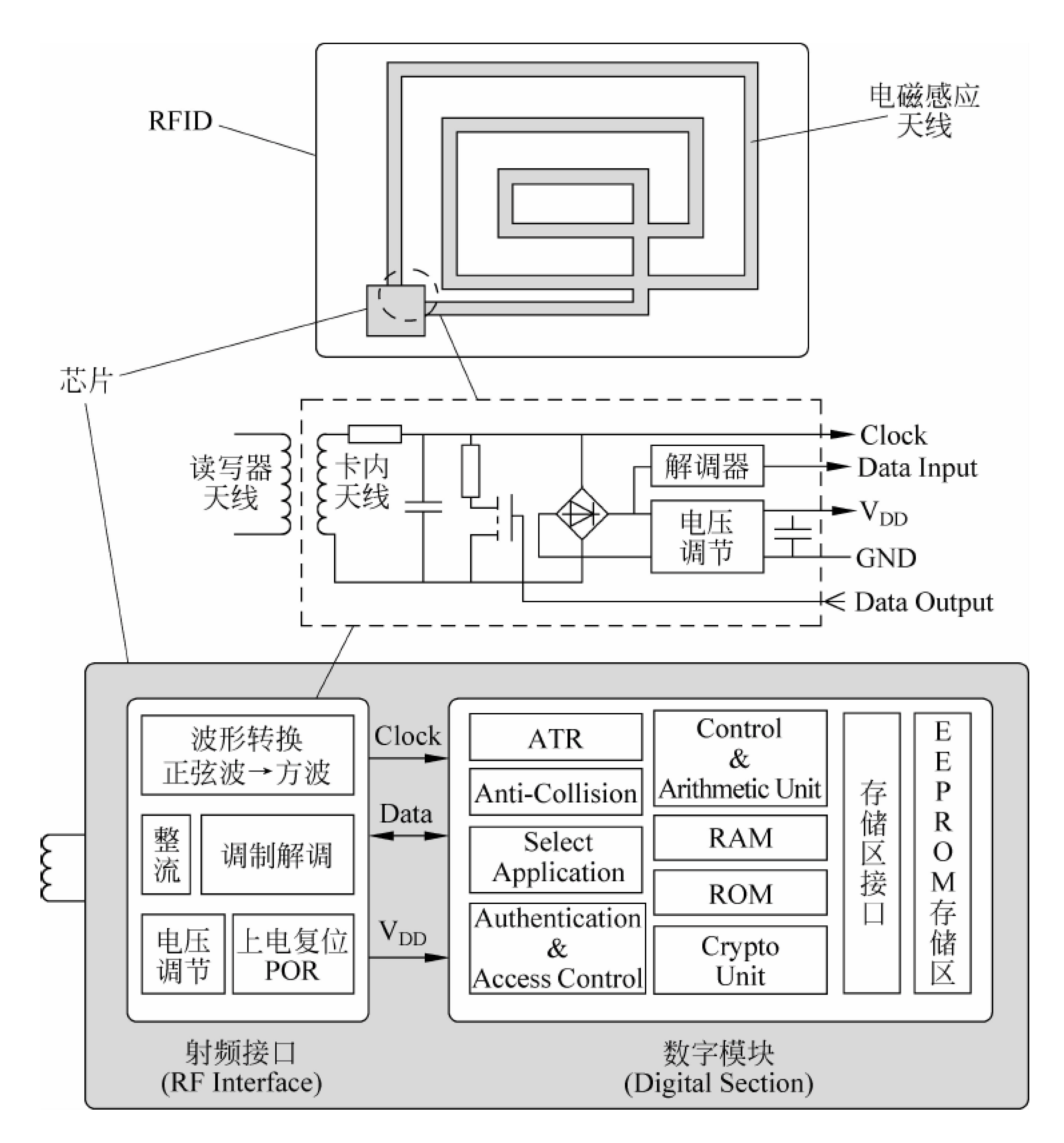


图 18.2 RFID 卡组成结构示意

RFID 卡的电气部分由一个感应天线和一块 RFID 芯片组成。数据保存期为 10 年,可改写 10 万次,读无限次。卡与读写器之间通信采用 DES 和 AES 加密算法,具有较高的保密性。

工作频率为 f_c=13.56 MHz;数据传送速率 f_c/128=106Kb/s;具有防冲突功能,同一时间可处理多张卡;读写距离小于 100mm;采用握手式半双工通信方式;每个数据块有 16b CRC 纠错,每字节进行奇偶校验;电感耦合式能量传递;芯片采用高速的 CMOS

EEPROM 工艺；空中接口的所有数据均进行加密；每一扇区有相互独立的密码；每张卡有 32b 全球唯一的序列号；每次会话过程通常小于 100ms；抗静电保护能力为 2kV。

如图 18.2 所示，卡的芯片由射频接口和数字模块两部分组成。

射频接口为 RFID 芯片内部各部分电路提供：工作时所需要的能量；上电复位（Power On Reset，POR）信号，使各部分电路同步启动工作；从载波中提取的时钟信号，供数字模块使用；发送和接收数据的调制与解调。

读写器对有效工作区域内的 RFID 卡发射激励信号（一组固定频率的电磁波），卡内的 LC 串联谐振电路产生共振，使电容内积累了电荷，在电容另一端的单向导通电子泵将电容内的电荷送到另一个电容内储存，当电荷达到 2V 时，即可作为电源为其他电路器件供电。

数字模块主要由如下部分组成。

（1）ATR（Answer to Request）模块：响应读写器发出的 Request all 命令。

（2）Anti-Collision 模块：若有多张卡同时在读写器天线的工作范围内时，此模块启动，读写器获取每一张卡的序列号，然后选定一张卡片。

（3）Select Application 模块：确认对卡片的选择。

（4）Authentication & Access Control 模块：完成卡片与读写器之间认证。

（5）Control & Arithmetic Unit（CAU）模块：相当于中央处理机单元。

（6）RAM：暂存 CAU 计算结果，可进一步保存到 EEPROM，或进行发送。

（7）ROM：固化卡片运行的程序指令。

（8）Crypto Unit：完成对数据的加密处理及密码保护。

如图 18.3 所示，读写器向卡传输信号时，射频载波频率 $f_c=13.56\text{MHz}$，采用同步时序、Modified Miller 编码方式，调制深度为 100% 幅移键控（Amplitude Shift Keying，ASK）；卡向读写器传送信号时，使用的副载波频率为 $f_s=f_c/16=847\text{kHz}$，采用 Manchester 编码，开关键控（On-Off Keying，OOK）调制信号。

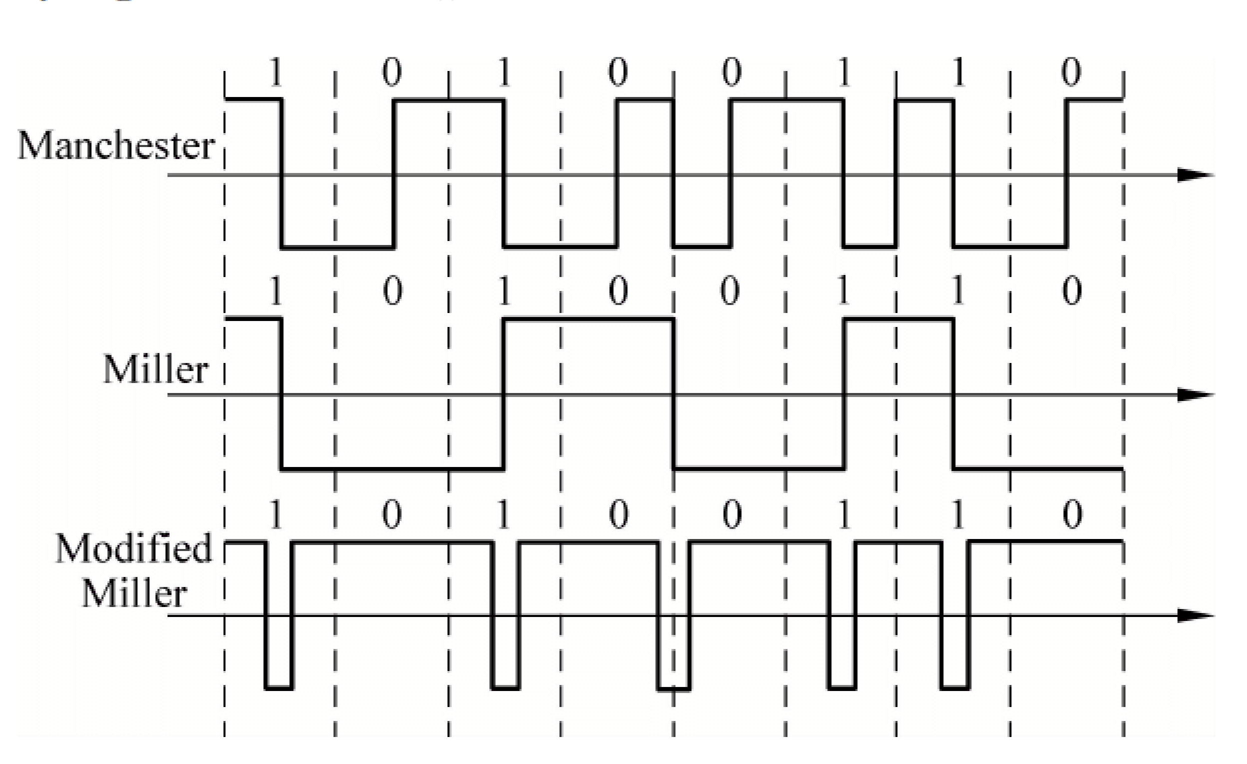


图 18.3　RFID 3 种编码方式波形比较

RFID 存储器访问方式如下。

如图 18.4 所示，EEPROM 总容量为 8Kb=1024B，分为 16 个扇区 0～15，每扇区 64B 被进一步分为 4 个块 0～3，每块 16B。

扇区 0 的块 0 为固化的厂商代码，保存卡序列号（SN）、容量和卡类型等信息，只读操作，不可改写，其他块可根据设定的规则进行独立的读写操作。

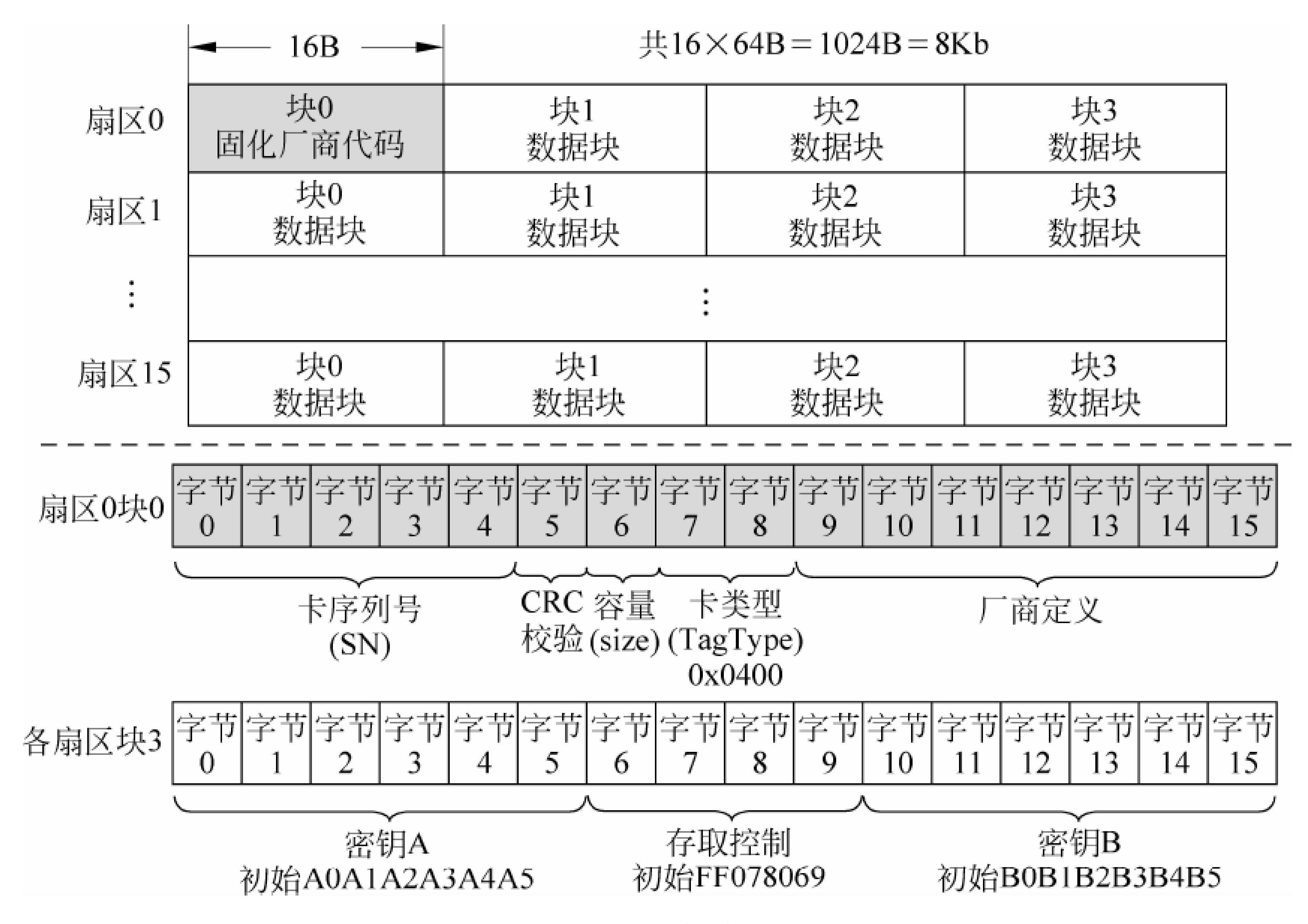


图 18.4 RFID 存储空间结构

如表 18.1～表 18.3 所示,每扇区的块 0～2 为数据块,块 3 为控制块,数据块的读写操作完全受对应扇区的控制块的权限控制,可以设置为不可读(指不能在空中接口传输)但可写(可修改、可删除)、可读且可写。每扇区配置两个独立的密钥,所有数据存取操作都应首先使用指定密钥通过相关认证后才可继续执行。

表 18.1 块 3 中 3 个控制位的位置

	bit0	bit1	bit2	bit3	bit4	bit5	bit6	bit7
字节 6	$\underline{\text{c1b0}}$	$\underline{\text{c1b1}}$	$\underline{\text{c1b2}}$	$\underline{\text{c1b3}}$	$\underline{\text{c2b0}}$	$\underline{\text{c2b1}}$	$\underline{\text{c2b2}}$	$\underline{\text{c2b3}}$
字节 7	$\underline{\text{c3b0}}$	$\underline{\text{c3b1}}$	$\underline{\text{c3b2}}$	$\underline{\text{c3b3}}$	c1b0	c1b1	c1b2	c1b3
字节 8	c2b0	c2b1	c2b2	c2b3	c3b0	c3b1	c3b2	c3b3
字节 9	备用	备用	备用	备用	备用	备用	备用	备用
控制位说明	↑块 0	↑块 1	↑块 2	↑块 3	↑块 0	↑块 1	↑块 2	↑块 3

说明:$cibj$ 表示块 j 的控制比特 i,$\underline{cibj}$ 表示该比特取反。

表 18.2 控制位对各扇区块 0～2 的存取控制

c1bj	c2bj	c3bj	读 (Read)	写 (Write)	加值 (Increment)	减值/传送/重置 (Decrement/Transfer/Restore)
0	0	0	keyA|B	keyA|B	keyA|B	keyA|B
0	1	0	keyA|B	Never	Never	Never
1	0	0	keyA|B	keyB	Never	Never
1	1	0	keyA|B	keyB	keyB	keyA|B
0	0	1	keyA|B	Never	Never	keyA|B
0	1	1	keyB	keyB	Never	Never
1	0	1	keyB	Never	Never	Never
1	1	1	Never	Never	Never	Never

说明:$cibj$ 表示块 j,j=0,1,2 的控制位 i,keyA|B 表示密钥 A 或 B,Never 表示不可操作。

表 18.3　控制位对各扇区块 3 的存取控制

c1b3	c2b3	c3b3	keyA		control		keyB	
			Read	Write	Read	Write	Read	Write
0	0	0	Never	keyA|B	keyA|B	Never	keyA|B	keyA|B
0	1	0	Never	Never	keyA|B	Never	keyA|B	Never
1	0	0	Never	keyB	keyA|B	Never	Never	keyB
1	1	0	Never	Never	keyA|B	Never	Never	Never
0	0	1	Never	keyA|B	keyA|B	keyA|B	keyA|B	keyA|B
0	1	1	Never	keyB	keyA|B	keyB	Never	keyB
1	0	1	Never	Never	keyA|B	keyB	Never	Never
1	1	1	Never	Never	keyA|B	Never	Never	Never

说明：c*i*b3 表示块 3 的控制位 *i*，keyA|B 表示密钥 A 或 B，Never 表示不可操作。

RFID 存取操作首先需要经过安全鉴别，防止非法操作或误操作。认证过程分为如下五步。

(1) 卡片向读写器发送随机数 R_B。

(2) 读写器收到 R_B 后向卡片发送一个令牌数据 TOKEN-A_B，其中包含了用读写器中存放的密钥加密后的 R_B 及读写器生成的随机数 R_A。

(3) 卡片收到 TOKEN-A_B 后，用卡中的密钥对 TOKEN-A_B 中的加密数据进行解密得到 R'_B，若 $R_B = R'_B$，鉴别通过，进行下一步。

(4) 卡片再用卡中存放的密钥对 R_A 加密后发送令牌 TOKEN-B_A 给读写器。

(5) 读写器解密得到 R'_A，若 $R_A = R'_A$，鉴别完成。

例如，块 3 中存取控制的 4B 的初始化值(厂商初始值)为 0xFF、0x07、0x80、0x69，则根据表 18.1 有 c1b0-c2b0-c3b0＝000，c1b1-c2b1-c3b1＝000，c1b2-c2b2-c3b2＝000，c1b3-c2b3-c3b3＝001。根据数据块存取控制规则(见表 18.2 中的第一种情况)，说明在初始状态下用密钥 A 或 B 验证后，可以对数据块 0～2 进行读、写、加值、减值、传送、重置等操作；根据控制块存取控制规则(见表 18.3 中的第五种情况)，用密钥 A 或 B 验证后，可以写密钥 A，可以读/写控制位，可读/写密码 B，但不能读密码 A。

芯片 EEPROM 中的数据块有两种应用方法：一种是用于一般的数据保存，直接进行读写；另一种是作为数值，可以进行初始化、加值、减值、读值的运算。应用系统配用相应的接口函数完成所需的功能。

18.1.3　GPS

1. GPS 原理

全球定位系统(Global Positioning System/Service，GPS)，又称卫星定位服务，是利用卫星为行人、设备、车辆、船舶和飞行器提供方位指示的服务系统，以确定所处位置的经度、纬度、高度、时间，并可据此计算运行方向和速度(如图 18.5 所示)。GPS 只是美国部署的卫星定位系统的名称，但其投入运行最早，目前为唯一全球服务的现役系统，常用 GPS 来指代卫星定位服务。

GPS 的用途很广，大到飞机、船舶、导弹的导航，小到路线指示、汽车防盗，位置信息都

是必不可少的。如今越来越多的手机、平板电脑等都配备了GPS模块,可以支持基于位置的服务(LBS),今后还会产生更多形式的终端、更多种类的服务,GPS技术也将发挥更大的作用。

GPS是一种特殊的无线通信系统,虽然卫星之间、卫星与地面设备之间采用数字无线传输技术,但目的并不在于设备间的通信,而是通过协议数据所携带的位置和时间信息,计算自身所处的方位。此外,GPS不但可以用来定位,还可以用来实现网络中不同设备间的时钟同步。

卫星导航定位系统的全球部署情况如下。

(1) 美国GPS:由美国于20世纪70年代初开始设计,1993年全部建成。

(2) 欧盟伽利略(Galileo):1999年由欧盟提出计划,准备发射30颗卫星。

(3) 俄罗斯格洛纳斯(Glonass):始于20世纪70年代,截至2009年1月30日,拥有18颗卫星的格洛纳斯信号已可覆盖俄罗斯全境。

(4) 中国北斗(BeiDou或Compass):2003年5月25日北斗一号卫星成功定点,我国启动了自主的导航卫星系统,2011年起已经可以在亚太地区商用。预计到2020年北斗系统将覆盖全球。

美国GPS系统由24颗绕极卫星组成,分布在6个轨道平面上,每个轨道平面4颗卫星,位于20 200km高空,轨道倾角为55°,绕地球一周为11h58min。位于地平线以上的卫星颗数随时间和地点的不同而不同,最少可见4颗,最多可见11颗(如图18.6所示)。

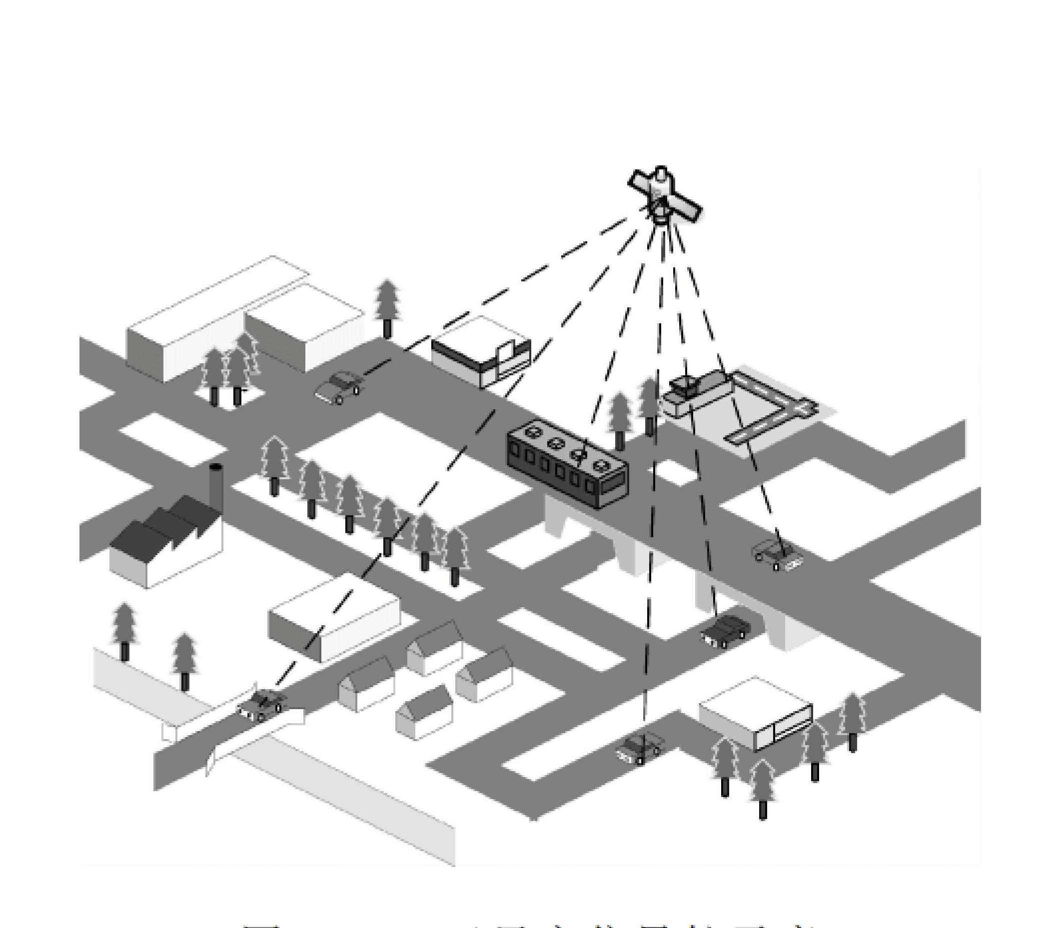


图18.5 卫星定位导航示意

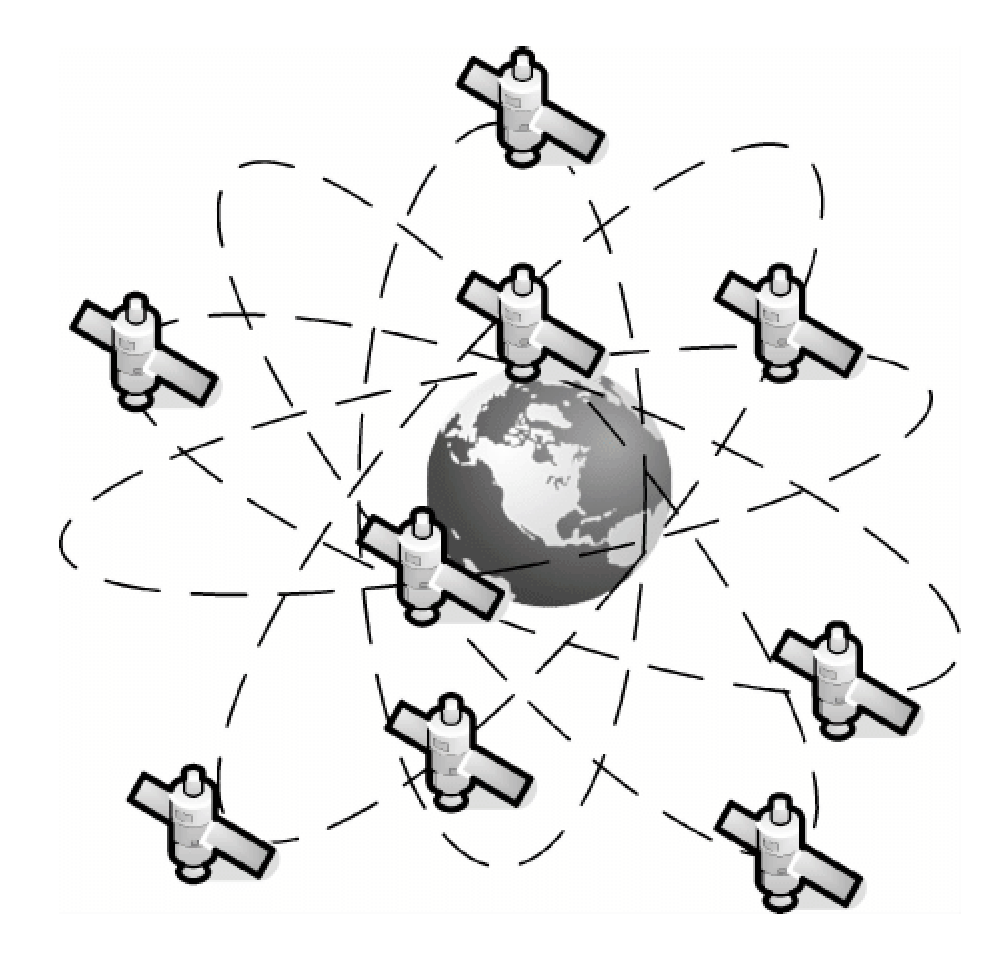


图18.6 GPS卫星分布示意

GPS地面控制系统由一个主控站、5个监测站和3个注入站所组成(如图18.7所示)。主控站位于美国科罗拉多州(Colorado)的Spring市。地面监测站负责收集由卫星传回的信息,经初步处理后,传送给主控站,计算卫星星历、相对距离、大气校正、钟差等数据,并通过注入站发送给卫星。

卫星时钟频率 $f_0=10.23\text{MHz}$,在L波段的两个指定载频上发送扩频信号,分别采用两种伪随机码调制方法:L1频段($154\times f_0=1575.42\text{MHz}$)为民用,波长 $\lambda_1=19.03\text{cm}$,采用低精度的C/A(Coarse/Acquisition)码,重复周期为1ms,发送频率为 $f_0\div 10=1.023\text{MHz}$,即码间距约1$\mu$s,等效于300m;L2频段($120\times f_0=1223.6\text{MHz}$)为军用,波长 $\lambda_1=24.42\text{cm}$,采

用 P/Y 码，P(Precise)码重复周期为 266.4 天，频率为 10.23MHz，即码间距为 0.1μs，等效于 30m；Y 码是在 P 码的基础上形成的，保密性能更佳。

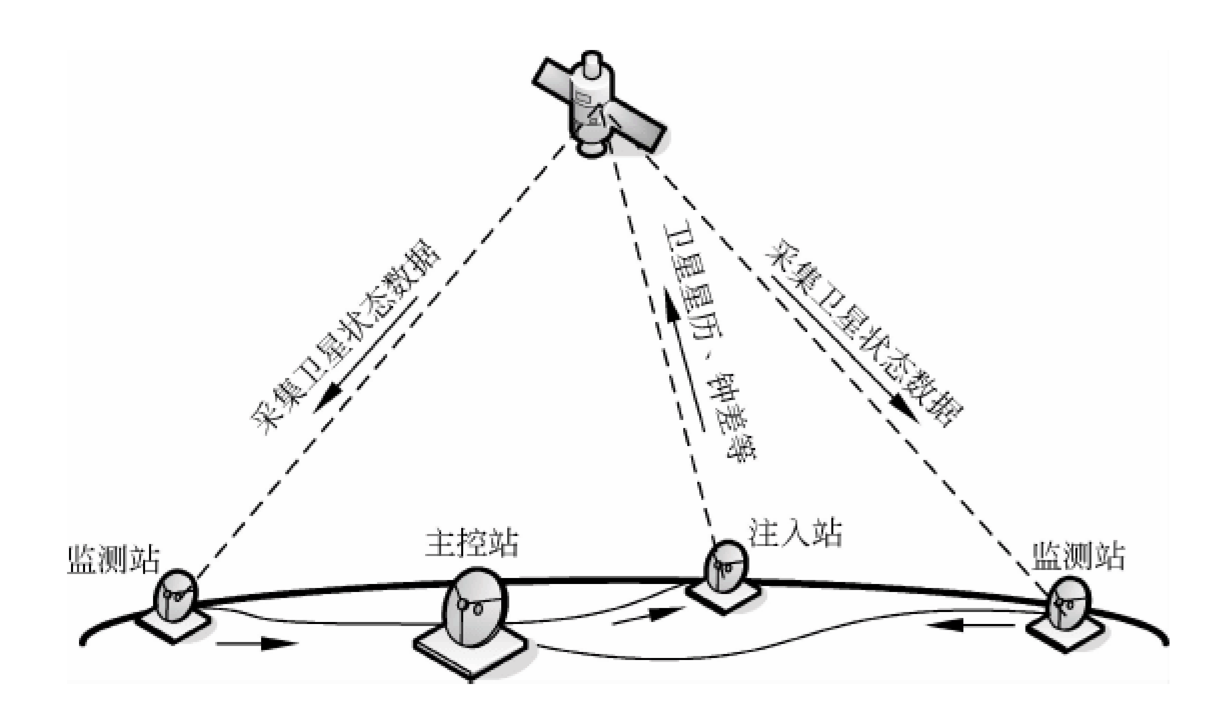


图 18.7　GPS 地面站系统示意

导航电文包括卫星星历、工作状况、时钟修正、电离层时延修正、大气折射修正等信息，以 50b/s 速率调制在载频上发射，采用不归零制(NRZ)二进制编码，可从卫星信号中解调出来。如图 18.8 所示，导航电文每个帧中包含 5 个子帧，每个子帧长 6s。前 3 帧各 10 个字，每 30s 重复一次，每小时更新一次；后两帧共 15 000b，各包含 25 页。导航电文中的内容主要有遥测码，转换码，第 1、2、3 数据块 5 部分，其中最重要的是星历数据。

在一个二维平面中，利用已知 3 点的坐标和距离(即半径)，可唯一确定一个点的位置，这就是三角定位的基本方法。如图 18.9 所示，设 3 个点坐标分别为(x_1,y_1)、(x_2,y_2)、(x_3,y_3)，到待测点(x,y)的距离分别为 r_1、r_2、r_3，则解以下方程组即可：

$$r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2},\quad i = 1,2,3$$

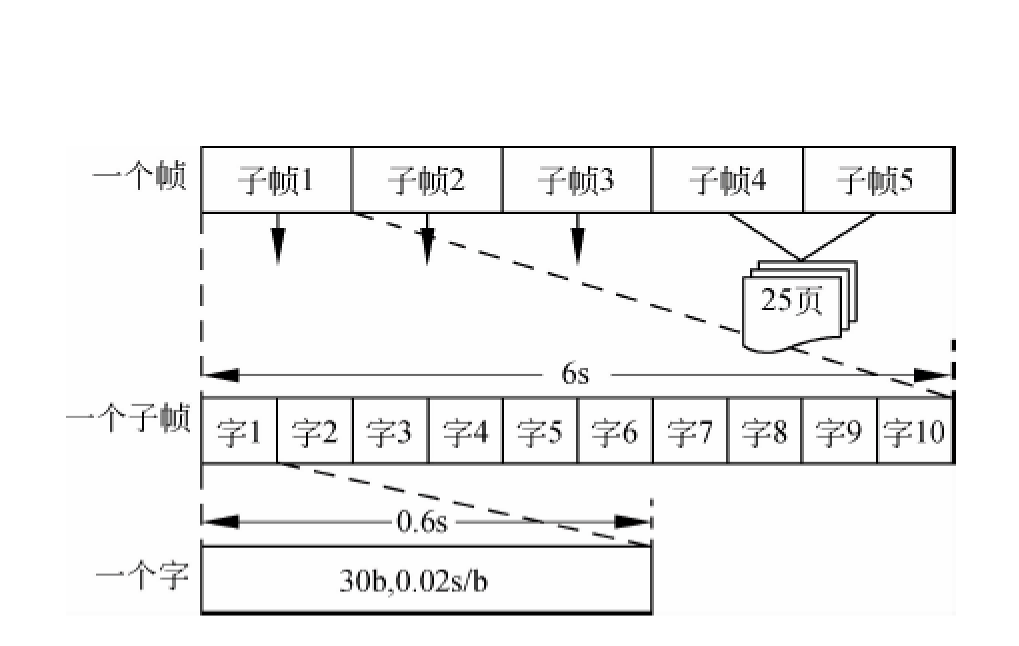


图 18.8　GPS 导航电文帧结构

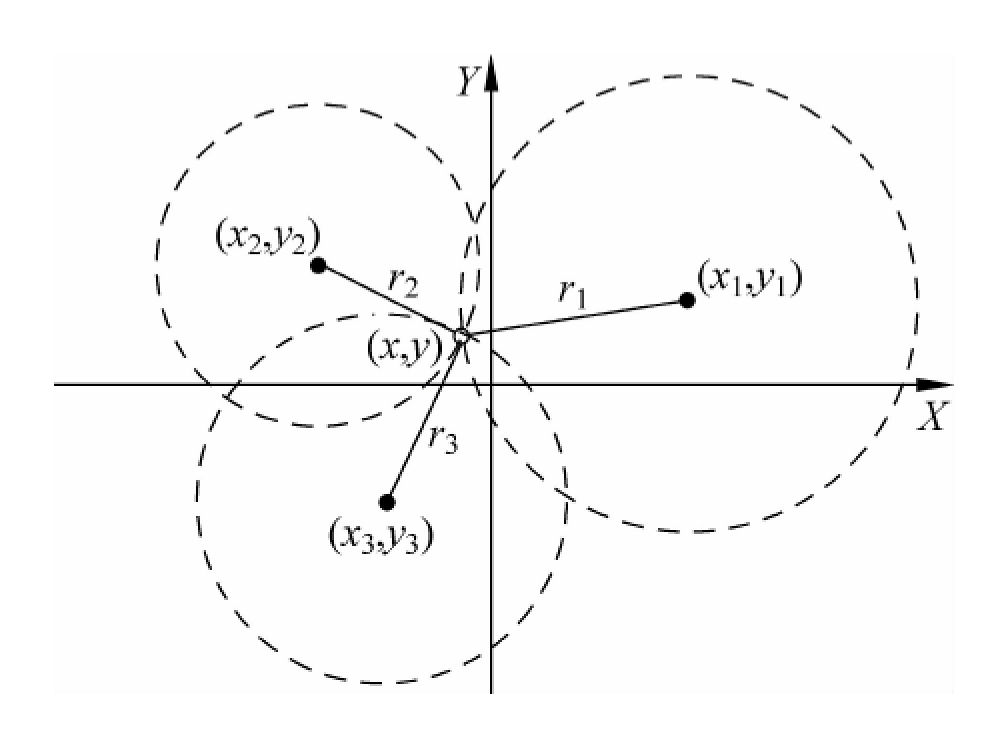


图 18.9　三角定位原理

可根据三角定位的原理，利用基站实现移动站点的定位，距离 r_i 可通过无线信号衰减率或时钟同步方式测算。

若为三维空间，如图 18.10 所示，同样可根据已知 3 点的坐标(x_i,y_i,z_i)和距离 r_i，唯一确定(x,y,z)点的位置。解以下三元方程组即可。

$$r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2},\quad i = 1,2,3$$

卫星定位即利用该 3 点定位的原理，依据是卫星所在的空间位置(经度、纬度、高度)坐标和信号传输时延计算出的距离(信号传输速度按光速 c 计)。

然而,由于各个卫星和地面接收终端的时钟不可能绝对相同(同步),必然存在误差,如果不加考虑,计算结果的位置偏离可能非常大,而且不可预测。因此,需要第四颗卫星的数据,共同计算加入时钟偏差因素的准确位置。

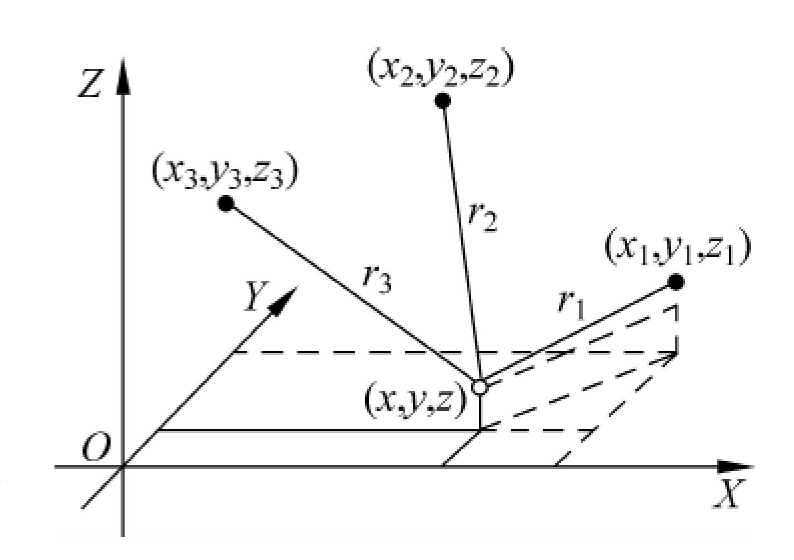


图 18.10　三点空间定位法原理

卫星的星历数据,包括空间位置和时钟偏差(clock bias)均由地面控制站统一注入,并通过导航电文发送给接收终端。如图 18.11 所示,设星历数据表示的(经度、纬度、高度)空间位置分别为(x_i,y_i,z_i),时钟偏差(钟差)分别为 Δ_{ti}。

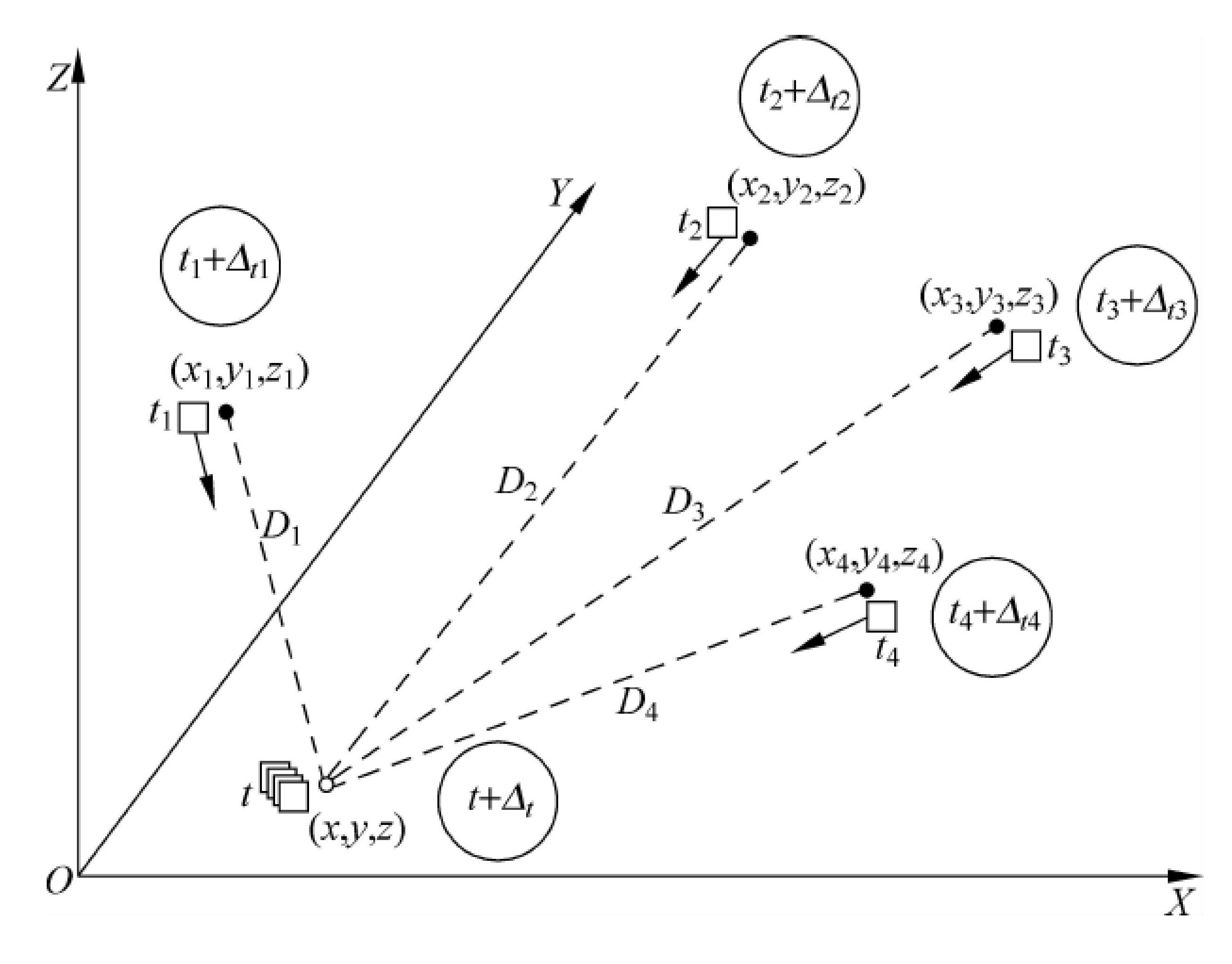


图 18.11　GPS 定位原理

那么,当导航电文由卫星同时发出,发出时卫星时钟分别为 t_i,则精确(标准)时钟应为 $t_i+\Delta_{ti}$。电文到达接收终端时,设终端时钟为 t,终端的钟差未知,为 Δ_t,则精确时钟为 $t+\Delta_t$。根据电文传输时间和3点定位原理,可计算卫星与接收终端之间的距离 D_i,列方程式如下。

$$\begin{cases} D_i = c((t+\Delta_t)-(t_i+\Delta_{ti})) = c(t-t_i)+c(\Delta_t-\Delta_{ti}) \\ D_i = \sqrt{(x_i-x)^2+(y_i-y)^2+(z_i-z)^2} \end{cases},\quad i=1,2,3,4$$

解方程组即可得(x,y,z)和 Δ_t。也可采用另一种方法,根据发送电文时的卫星时钟和电文到达时间计算距离 d_i,因为并非两者的真实距离,所以称为**伪距**(psudo-range,PR),即上式中的 $c(t-t_i)$部分,从而方程式改写为

$$d_i = \sqrt{(x_i-x)^2+(y_i-y)^2+(z_i-z)^2}+c(\Delta_{ti}-\Delta_t),\quad i=1,2,3,4$$

虽然 GPS 卫星均采用高精度原子钟,但与标准时间的偏差和漂移仍有 0.1～1ms 之多,由此引起的等效误差将达到 30～300km,可谓失之毫厘,差之千里。明显可见,方程式中任何一项的误差都将直接影响到最终的定位准确性,包括星历数据本身的误差、大气层(电离层、对流层)对信号传输造成的影响(折射、多路径效应等)。另外,由于卫星在太空高速运动的特性,相对论效应也会产生一定的影响。

美国军方为了限制非特许用户利用 GPS 进行高精度定位,采用一种特殊的故意降低系

统精度的方法，称为 SA 政策，包括降低广播星历精度的 ε 技术和在卫星基本频率上附加一随机抖动的 δ 技术。实施 SA 政策后，SA 误差成为影响 GPS 定位误差的最主要因素。虽然美国在 2000 年 5 月 1 日取消了 SA，但必要时，可以恢复或采用类似的干扰技术。

采用差分 GPS(DGPS)可进一步提高定位精度。根据差分 GPS 基准站发送的信息方式可分为 3 类：位置差分、伪距差分和相位差分。这 3 类方式的工作原理是相同的，都是由基准站发送修正值，由终端接收并对其测量结果进行修正，以获得精确的定位结果；不同的是，发送修正值的具体内容不一样，其差分定位精度也不同。

2. 北斗系统

2000 年 10 月 31 日、12 月 21 日，2003 年 5 月 25 日，北斗一号 01、02 星和备用卫星在西昌卫星发射中心发射升空，标志着我国成功建立了自主的北斗导航试验卫星系统。北斗系统的指挥机和终端之间可以通过卫星进行双向交流，成为一大技术特色。

北斗卫星导航定位系统是中国自行研制开发的区域性有源三维卫星定位与通信系统(CNSS)，由两颗北斗一号地球静止轨道工作卫星和一颗备用卫星组成，由地面控制中心进行控制，是覆盖中国本土的区域导航系统，范围为东经 70°～140°，北纬 5°～55°，在地球赤道平面上的二颗地球同步卫星的赤道角距约 60°。北斗系统定位精度约几十米，授时精度约 100ns，略低于 GPS 系统的 6～12m 和 20ns。

北斗卫星导航定位系统的基本工作原理是双星定位：以两颗在轨卫星的已知坐标为圆心，各以测定的卫星至用户终端的距离为半径，形成两个球面，用户终端将位于这两个球面交线的圆弧上。地面中心站配有电子高程(高度)地图，提供一个以地心为球心，以球心至地球表面高度为半径的非均匀球面。用数学方法求解圆弧与地球表面的交点即可获得用户的位置。

由于北斗在定位时需要用户终端向定位卫星发送定位信号，由信号到达定位卫星时间的差值计算用户位置，所以被称为有源定位。但是，用户终端发送信息可能暴露其位置，在军事上处于不利局面。

未来的第二代北斗卫星导航系统将由 5 颗静止地球轨道(GEO)卫星和 30 颗非静止轨道卫星组成，提供两种服务方式：开放服务和授权服务。开放服务是在服务区免费提供定位、测速和授时服务，定位精度为 10m，授时精度为 50ns，测速精度达到 0.2m/s。授权服务是向授权用户提供更安全的定位、测速、授时和通信服务以及系统信息。

第二代北斗卫星导航系统的基本工作原理与 GPS 类似。为了保持地面控制系统各站之间，以及地面站与卫星之间的时间同步，通过站间和星地时间比对观测与处理，使时间同步精度更高。

18.1.4 泛在计算

泛在计算(Ubiquitous Computing)或称**普适计算**是关于计算机和网络应用向深度、广度延伸的思想。普及计算基于计算机的智能、互联网的联通两方面的能力，使信息化、智能化到达生活的各个层面、各个角落。

在泛在计算环境中，各种具有计算和联网功能的设备随处可见，从而使计算能力像水和电一样，随处可用、按需索取。如图 18.12 所示，人类活动所需要的计算能力的满足，从安装在专用机房的大型机，延伸到人们伸手可及的地方。或许 20 年前人们还难以想象昂贵的、

神奇的、高科技的 CPU 会装到一个廉价的电子产品中,而如今已经变成现实。

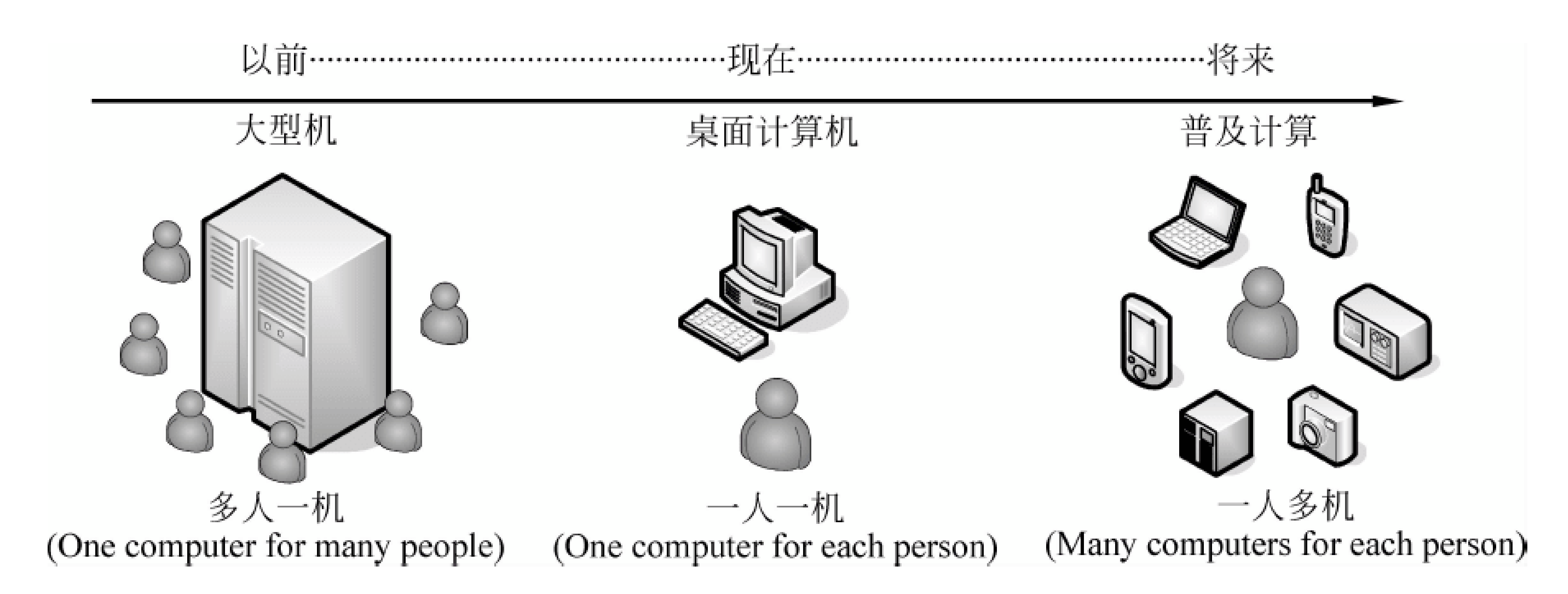


图 18.12 泛在计算发展历程示意

物联网技术的运用推进了泛在计算的发展。物联网使非智能物体智能化,智能体尺寸变小、数量增大、无处不在,成为智能灰尘(intelligent dust;smart dust),达到人与机器、物品、环境的协调,在满足人类各种需求的目标下,实现自然的、优雅的计算(invisible computing)。

18.2 云计算

18.2.1 网格计算

网格(grid)技术提出于 20 世纪 90 年代,全称为**全球网格**(Great Global Grid,GGG)。grid 在英语中的本意就是一种网络,如电网、自来水管网、燃气输送网、蒸汽供热网等。之所以提出网格的概念,归根结底是对 Internet 网络服务现状的不满。

Internet 资源(resources)可以用 CFS 来概括:新闻、评论、游记、小说、论文等构成内容(Content);电子邮件、网络论坛、即时通信、电子支付、网上交易、视频点播等称为功能(Functions);主机系统、数据存储、在线软件等属于服务(Services)。这些资源实际上分散在各个国家和地区的各种网站及其不同的计算机上,要获取这些资源首先要了解资源所在的位置(URL),其次要知道资源的类型以便确定相对应的工具(协议),然后要适应各不相同的访问方式(注册、登录、访问流程等),最后才能谈得上享用资源。有些需求还要组合各种资源才能满足,非常辛苦。

通过搜索引擎(search engine)检索信息是了解 Internet 资源分布的有效方法。但传统的搜索引擎技术存在较大不足,以至于信息越多,找到所需信息的难度就越高,与“内容体现价值”的常理完全背道而驰。此外,基于关键字(keyword)的搜索技术设定了较高的技术门槛,使搜索结果取决于使用者的“语文水平”的高低,不符合技术公平服务的初衷。此外,搜索一般仅限于字和词的匹配,与语义、语境无关,使得搜索引擎更像是一把霰弹枪,而非精确击中目标的狙击枪。例如搜索“长城”,结果包含了万里长城、葡萄酒、汽车、电脑、证券、基金、公司、宽带接入、润滑油、歌曲、书籍、图片……要从包罗万象的 5000 多万条信息中获取真正想要的东西,无异于大海捞针。

网格是借鉴电力网的服务模式提出的,网格的最终目的是希望用户在使用 Internet 解决问题时像使用电力一样方便,用户不用去考虑得到的服务来自哪个地理位置(发电站),由

什么样的计算设施提供(输电网),采用何种技术(变电所),只需通过简单操作即可达到目的(打开电灯开关)。这样的例子还有很多,例如,拧开水龙头喝水,点燃液化气灶烧饭,开启暖气片取暖,接上电话机打电话等。也就是说,网格给最终的使用者提供的是一种通用的计算能力。

所以,网格试图利用 Internet 把分布于不同地域的计算机、数据库、存储器、软件、文件等资源连成整体,就像一台超级计算机一样为用户提供一体化信息服务,其核心思想可以简单表述为“Internet 就是一台计算机”。网格技术充分实现资源共享,具有低成本、高效率、易使用等优点。

网格技术最著名的经典案例莫过于寻找外星人的 SETI@home 计划了,最能够体现网格的思想和作用。在茫茫宇宙中寻找可能的同类一直是人类无法泯灭的梦想,方法是用分布在世界各地的射电望远镜采集电磁波信号,经过分析搜寻外星人智能化活动的蛛丝马迹(如包含信息的无线电波)。但海量数据的分析工作量十分巨大,况且对分析的结果无法预计,成功的希望也非常渺茫,用昂贵的大型计算机来做无休止的处理显然不是理智的做法。可行的方法是运用网格计算,在 Internet 上招募志愿者计算机,联网的计算机事先下载安装一个分析软件,类似屏幕保护程序,当计算机空闲时,程序激活开始工作,自动从数据中心下载数据并进行分析,然后将结果报告给数据中心。数百万台个人计算机提供的计算能力是惊人的,却几乎是零成本的。

与 WWW 类似,GGG 同样体现的是瘦客户机的思想,只是更进一步,网格将更多的技术细节隐藏起来,而使用户操作更加便捷。

由于网格需要良好的管理来维护正常的运行,其组成结构及资源调控必然相当复杂,需要解决的问题也很多。因为网格所关心的问题不再是文件交换,而是如何直接访问计算机、软件、数据和其他资源,且资源是动态变化的,这就要求网格具备解决资源与任务的分配和调度、安全传输与通信实时性保障、人与系统以及人与人之间的交互等能力。因此,建立网格技术统一的国际标准显得十分必要和关键。

网格研究最初的目标是希望能够将超级计算机连接成为一个可远程控制的元计算机系统(meta computers),如今这一目标已经深化为建立大规模计算和数据处理的通用基础支撑结构。开放网格服务结构(Openning Grid Service Architecture,OGSA)是 Global Grid Forum4 的重要标准建议,是一种最有影响力的网格体系结构,被称为下一代的网格结构。Globus 和 Web Service 是 OGSA 的两大支撑技术。OGSA 的目的就是要将 Grid 的一些功能融合到 Web Service 这个框架中。OGSA 是面向服务的结构,将所有事务都表示成一个 Grid 服务,计算资源、存储资源、网络资源、软件资源、数据资源等都是服务,所有的服务都联系对应的接口,所以,OGSA 被称为以服务为中心的“服务结构”,通过标准的接口和协议支持创建、终止、管理和开发透明的服务。

Globus 采用了以下协议。

(1) 基于 LDAP 的网格资源信息协议(Grid Resource Information Protocol,GRIP),用于定义一个标准的资源信息协议和相关的信息模型。相关的软件级的网格资源注册协议(Grid Resource Registration Protocol,GRRP)和网格索引信息服务器用于注册资源。

(2) 基于 HTTP 的网格资源访问与管理协议(Grid Resource Access and Management,GRAM)用于分配计算资源和监控这些资源上进行的操作。

(3) 网格文件传输协议(Grid FTP)是FTP的扩展版本。扩展部分包括运输层安全协议的使用,部分文件传输、高速传输中的并行机制管理。

Globus为每个协议定义了客户端、Java API及SDK,同时也为每个协议提供服务器端的SDK,以便使不同资源可以集成到网格中。如网格资源信息服务(Grid Resource Information Service,GRIS)实现服务器端LDAP功能;通用安全服务(Generic Security Services,GSS)API用于获取、转发和校验证书。

Web Service是一种标准化的访问网络应用的技术框架,XML标准相关工作是Web Service的基础。

Web Service包括几个比较重要的协议:简单对象访问协议(Simple Object Access Protocol,SOAP),Web服务描述语言(Web Service Description Language,WSDL),统一描述、发现与集成(Universal Description,Discovery & Integration,UDDI)和WS-Inspection。

网格是建立在Internet和Web基础上的,并非替代关系。Internet实现了信息共享,但离应用层面上的互连互通还有很大距离,联网计算机的使用也远不如电话那么方便,而网格技术将可能使之成为现实。

有观点认为,网格是在非集中控制的环境中协同使用资源,使用标准的、开放的和通用的协议和接口,提供非平凡的服务。

也有观点认为,网格是一种框架,把计算机软硬件等资源放在这个框架中,能够产生群体效应,能够完成所有的个体以及个体的集合不能完成的工作。

还有观点认为,应该把网格和网格技术分开来看待。网格技术是一种概念和思想,其本质是资源共享及协同工作,一切符合这个思想、能够完成传统技术不能完成的工作的技术都可以称为网格技术;网格则是一种基础设施,因为要建立在资源充分共享的基础上,所以网格要整合很多异构的资源和不同的信息系统,就需要依赖公共标准。网格标准应当是开放的,网格接口尽量简单,便于应用开发,而用户不必关心网格内部和底层实现的复杂性,所看到的网格是一个整体,完成一致的功能。

之所以存在不同的观点,一方面说明网格技术仍然需要一个不断完善和稳固的过程,另一方面说明了网格技术的适用面十分广阔,灵活性强,可以开拓不同的网格应用。但不管如何定义网格概念,有两个方面已经成为共识:网格是对Internet上各种资源的汇聚,以发挥巨大的累加效应;网格技术需要建立开放而统一的标准,便于更多有效资源加入进来。

根据网格技术和应用目标,可分为两种主要类型:专业网格和通用网格。

如图18.13所示为专业网格的工作原理。较有代表性的专业网格有计算网格、数据网格、存储网格,面向高能物理、生物、化学、天文、气象等领域。

如图18.14所示为通用网格的体系结构,设备之间可透过标准化的服务定义发现和使用各种服务。

可以看出通用网格与专业网格仅有很微妙的差别,事实上,在许多情况下,两者之间并没有严格的界限,是可以相互渗透、相互结合的。如果能够达到相互的完全融合,对于网格的发展无疑十分有益。

通用网格不但可以应用于Internet大环境中,而且可在很多局部的应用环境中发挥作用。例如:

(1) 召开一次会议,与会者是作为个体临时加入会议这个小环境,各个计算机设备具有

不同的结构、配置、性能和功能，利用网格服务发现技术可以将这些资源汇聚起来，演讲者可以直接将 PPT 输出给投影机，会议资料可以同步给所有人或从共享的打印机中打印出来，与会者可以擦写同一块白板，合作修改一个文档或设计稿，甚至随时引用其他人的某一段统计数据。

（2）救灾系统也可成为网格技术的用武之地。来自各地、各行业的专业或业余救灾队伍必须协调工作，所有资源应当物尽其用，及时出现在最能发挥作用的地方。运输车辆、医疗人员、后勤物资、生命探测仪、挖掘机、抢险人员等，采用网格技术可定义每个团队或个体的能力、数量、位置、持续工作时间等关键属性，并根据需求匹配最佳资源，进行合理调度。

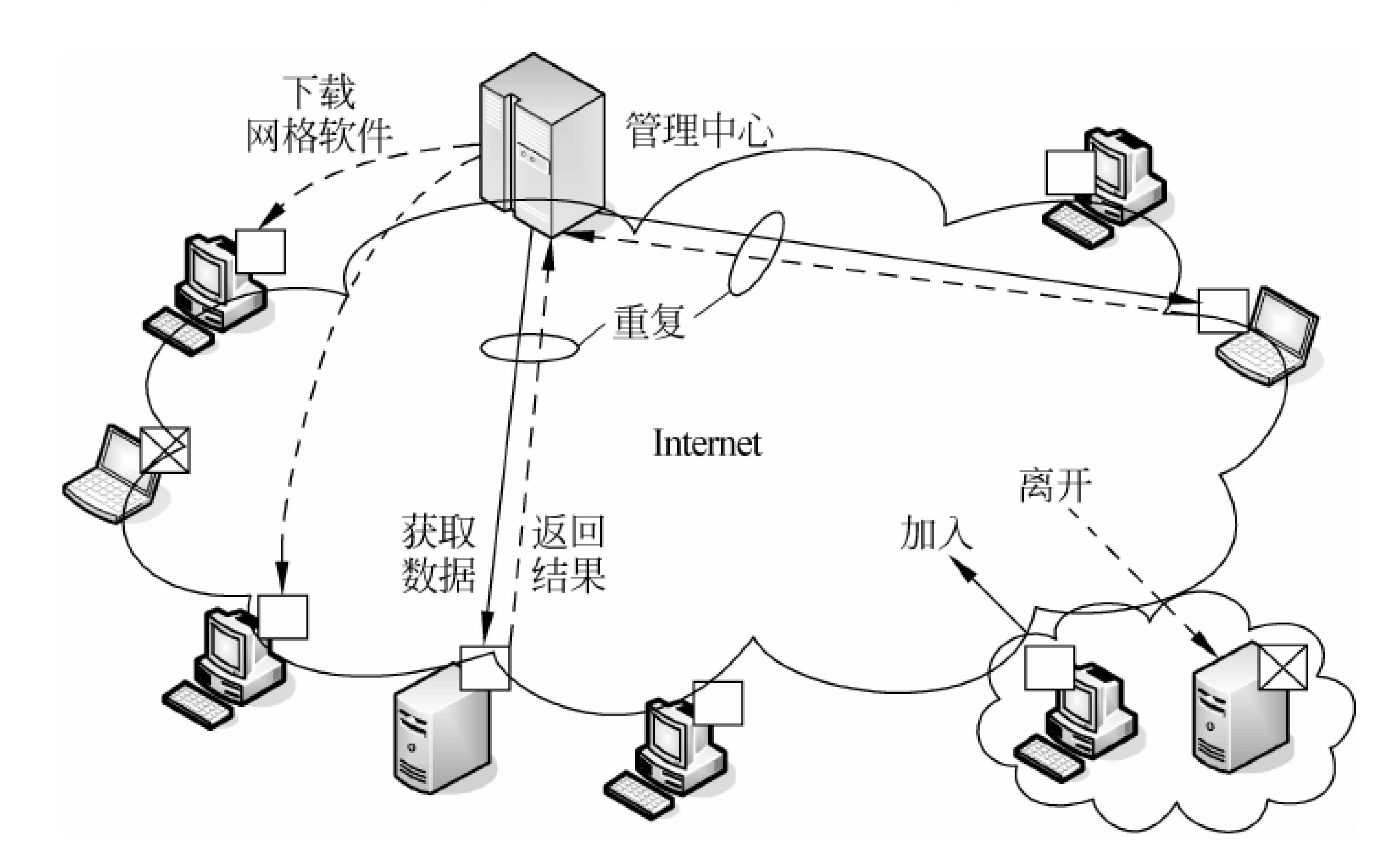


图 18.13　专业网格工作原理

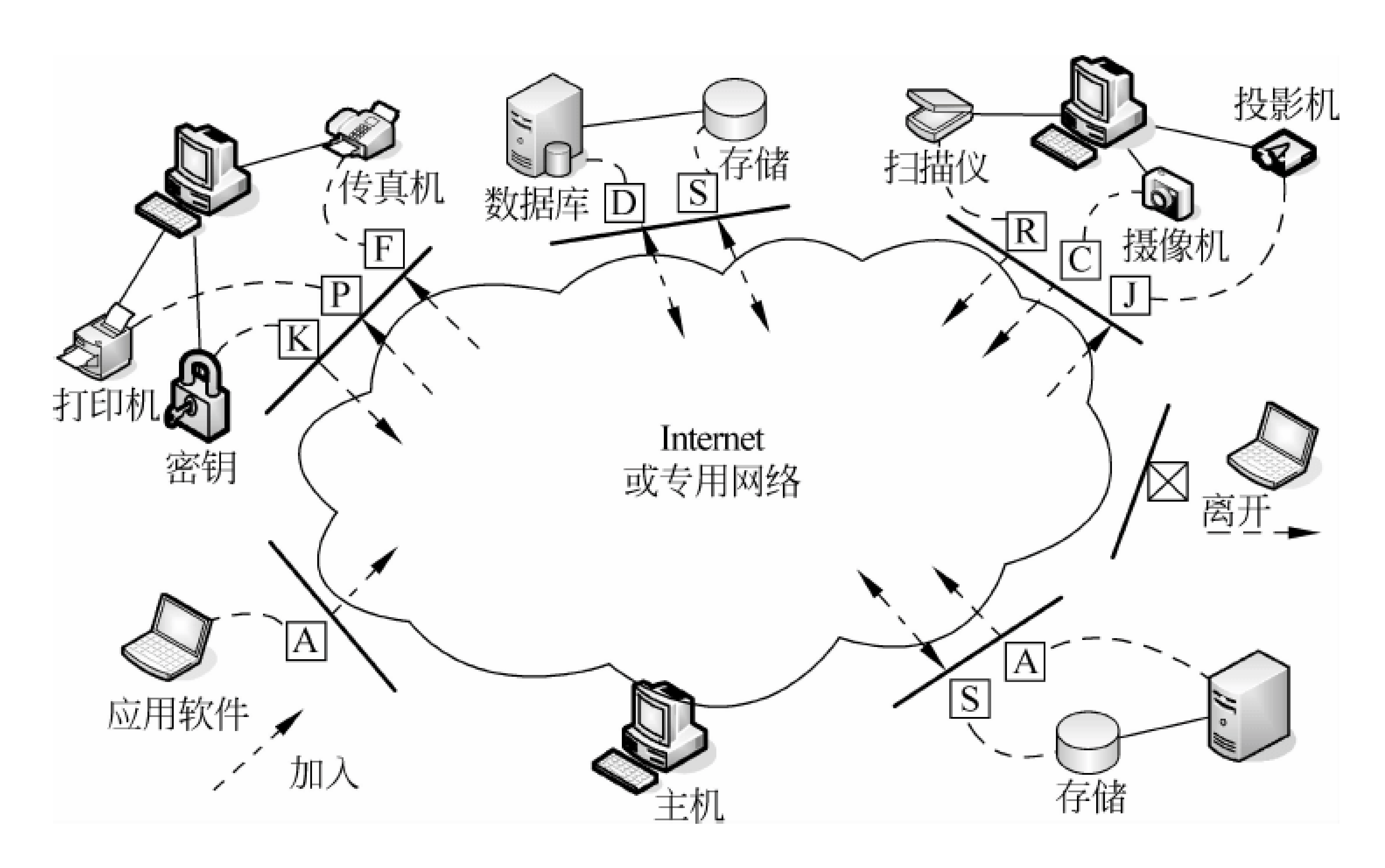


图 18.14　通用网格工作原理

18.2.2　云计算原理

云计算（Cloud Computing）概念的提出体现了人们对于 Internet 理想服务方式的追求。回顾计算机和网络的发展历程（如图 18.15 所示），可以发现一条计算模式改进的轨迹：从离线到联网，从孤立到合作，从独占到共享，从简单到复杂，从集中到分布，计算能力不断提

高。在一次次的进步中,从最初的瘦客户机(简单终端)到瘦服务器(C/S),再到瘦客户机(B/S),最终似乎又回到简单终端上,可见经历了螺旋式上升的发展曲线。

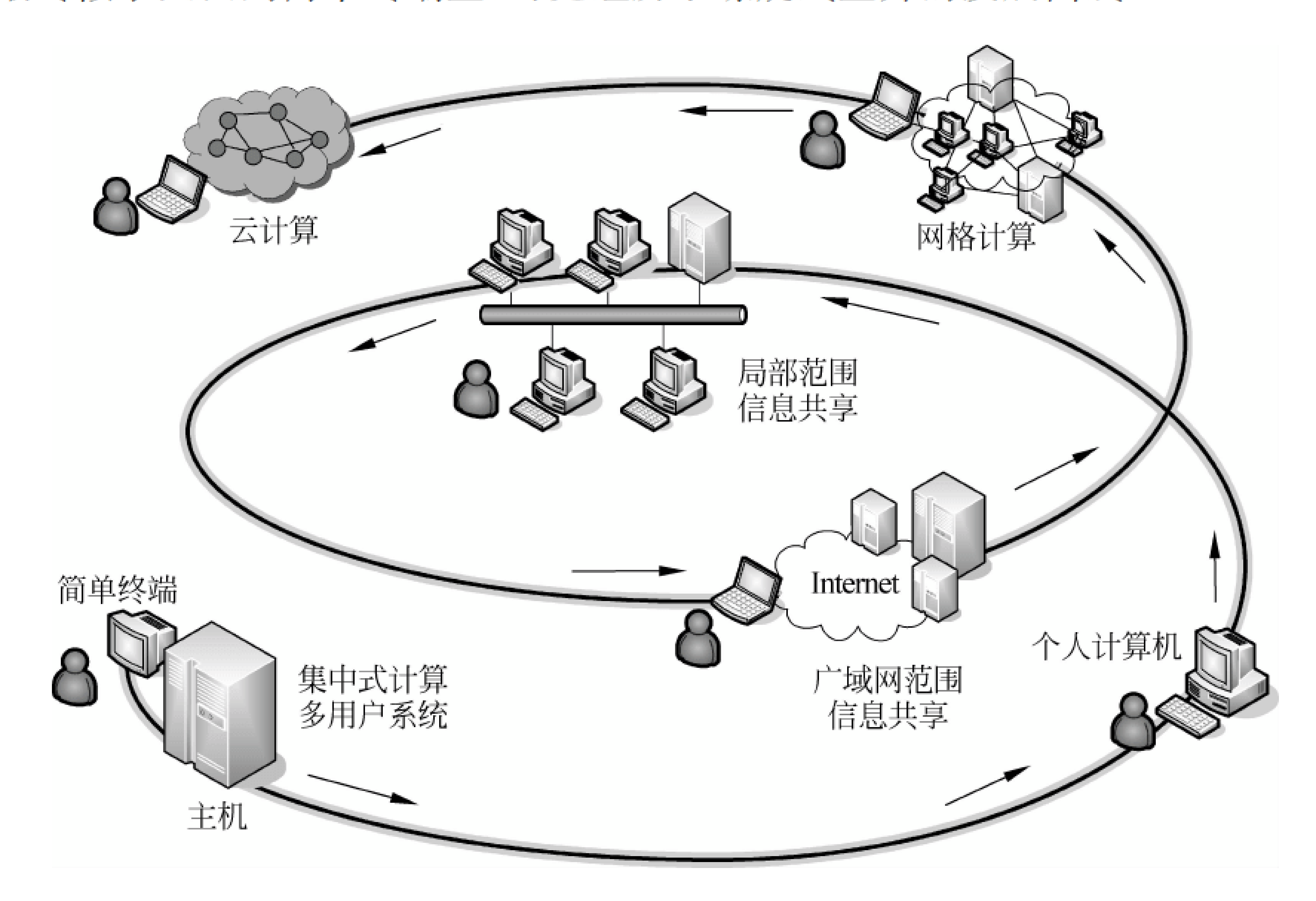


图 18.15　云计算技术发展轨迹示意

从中可以体会到云计算的含义:用户使用各种简单的终端,就可随时随地获取所需的服务,而不用关心网络的具体结构和技术,就像面对一朵看不到内部的云,这朵云就是一台无所不能的超级计算机。

在云计算之前,当 Web 服务逐步成为主流的信息访问方式后,**应用软件服务**(Application Service Provider,ASP)的模式就已经出现,个人和企业使用商用软件有了第二种选择:即租即用,而不是一次性购买使用权,但 ASP 只局限于极少数专业软件服务,不同于云计算的普遍服务。互联网数据中心(Internet Data Centre,IDC)则主要提供网站主机托管业务,包括带宽租用(物理托管)和虚拟主机等服务,可与域名服务、软件开发等打包,这与云计算面向个人用户的目标完全不同,但性能优异、管理完善的 IDC 可以成为云计算的重要物理基础设施。对等网络(Peer-to-Peer,P2P)实现了去中心化思想,让用户既是服务的需求者,也是服务的提供者,充分发掘和调动了分布在用户端的存储资源、带宽资源和计算资源,所以 P2P 技术也是云计算可以有效运用的手段之一。

云计算的理念在网格技术中已经被提出过,所以可以被视为网格的进一步发展(也有观点否认这一点)。网格计算比较强调体系结构的建立,研究服务的定义、服务的发现和服务的调用,而云计算则注重最终的表现,更多以用户的立场来看待问题。另一个差别在于网格计算提供的服务一般较为单一,例如完成一种并行计算任务、打印一个文档等,而云计算的服务综合性较强,为满足某个需求,通过结合一系列的网络服务来完成,例如从制定旅行计划,到购买机票、预订宾馆等,然后一揽子交付给用户。

理解云计算,可以从以下基本特征入手。

(1) 采用简单的网络终端,只需要一个标准浏览器或简单的客户端软件,甚至不需要复杂的操作系统(只需精简的嵌入式操作系统),几乎不需要安装各种应用软件(即所谓 no

software)，包括文档编辑、图像处理、文件压缩等均由网络服务提供，用户只需点击一项操作，云计算就会调用相应的程序来完成。文件系统应有的各项功能也由云计算提供，文件被保存在云存储中，用户通过一条虚拟的文件目录路径来访问虚拟的存储空间。

(2) 以用户为中心是云计算的指导思想。一旦一名用户接入云中，整个互联网就会围绕其需求展开，调动所有相关的设备、数据、应用程序、服务等，这些资源可以来自专业服务器，也可以来自其他用户的计算机。

(3) 任务驱动意味着软件工具是什么不再显得重要，焦点在于能够完成任务以及是否高效率、低成本。

(4) 易于访问是云计算追求的目标之一，使用户不再受到终端软硬件、数据来源以及网络操作能力等制约，而且支持强大的移动性，在任何时候，任何地点，用任何网络终端都可以获得同样的服务效果。

(5) 由于服务资源具有动态变化的特性，例如一台提供服务的计算机突然离线、一个应用程序站点临时失效，为了保证服务质量的始终如一和数据的安全可靠，云计算应具备智能化的资源管理和协调能力。

云计算技术包括云架构、云调度、云管理、云存储等方面，构成各种公共云或私有云服务提供给 Internet 用户。云服务的实现有以下 3 种主要方式。

(1) **软件即服务**(Software as a Service，SaaS)。SaaS 是一种基本的云服务，模式与 ASP 类似，但架构更开放，服务普及性更强。一个 SaaS 云服务提供商的背后是各种云服务应用软件的开发商，是对传统的开发者卖软件商业模式的颠覆。

(2) **平台即服务**(Platform as a Service，PaaS)。PaaS 是 SaaS 的一种变化，提供的云服务是开发环境。开发者利用 PaaS 云服务平台上的软件结构单元来创建自己的应用，并可进一步把创建好的应用以云服务的方式提供给用户。PaaS 可有效降低开发难度和成本，缩短开发周期，开发应用的过程就是构造云服务的过程，技术设计和商业计划有机结合在一起，而开发的成果反过来丰富了云服务。

(3) **设施即服务**(Intrastructure as a Service，IaaS)。IaaS 是以服务的形式，向用户提供服务器、存储、安全防范、数据库和其他网络基础设施。用户不必为了建立信息系统而添置大量的硬件和系统软件，只需租用虚拟化的环境，例如云存储、云安全，就可以快速建立自己的网络体系。

云计算尚存在许多问题亟待解决，首当其冲是技术规范的标准化。各自为政的局面只会导致用户无所适从。此外，数据安全、隐私保护、联网依赖、带宽限制等都是制约云计算发展的不利因素。

18.3　移动计算

18.3.1　移动计算原理

移动计算(Mobile Computing)泛指使用移动终端进行网络应用的活动。

移动计算是相对于固定位置的信息化应用，例如桌面计算机，或受到网络线、电源线牵制而动弹不得的便携计算机而言的。所以，用于移动计算的终端的特点一定是要能够自由

移动。根据这一原则,移动计算终端必须采用无线通信技术、轻便而续航能力强。对移动计算终端的其他要求可能包括性能优良、安全可靠、操作便捷、运行流畅、系统开放、设计美观。

适用于移动计算的无线通信技术有无线以太网 WLAN、2G/3G/4G 移动通信系统、蓝牙(BT)或近场通信(NFC)等。GPS 和 RFID 虽然也采用无线通信技术,可是并不能用来上网。从无线通信支撑移动计算的不同应用类型的角度看,有两种不同的技术构造方式。

(1) 准移动方式。移动计算终端可以在一定范围内移动,但不能离开当前无线通信基站的信号覆盖范围,一旦超过信号区域边界,连接就会中断。例如采用 WLAN 联网情况下,即使从正在连接的 AP 移动到另一个 AP 的范围,如果不采用 Mobile IP 等无缝切换技术,应用也会因此终止,需要重新开始。相对而言,基于 TCP 的应用实现无缝切换更为困难,而只要保持 IP 地址的连续性,基于 UDP 的应用容易实现接续。

(2) 全移动方式。移动计算终端可以随意移动,甚至允许以很高的速度大范围移动。实现完全的移动计算,需要无线通信网络的全覆盖、信道层面的无缝切换、高层协议逻辑连接的平滑过渡。采用移动通信网络可以满足前两个条件,而网络协议需要进行相应的配合。

移动计算终端类型很多,如笔记本电脑、平板电脑、手机(尤其是智能手机)、PDA、上网本、电子阅读器等。主流的移动计算操作系统有 Apple 的 iOS、Google 的 Android 和 MS 的 Windows 系列,其中 Android 的开放性最好。

移动计算与云计算技术有非常密切的、相互依存的关系。移动计算终端一般不具有台式计算机的超高性能,而是一种轻计算模式,属于瘦客户端类型,所以需要依赖网络(后台)的强大计算能力。云计算平台正具有这种能力。反过来,云计算平台的运算资源、应用资源、内容资源和存储资源,需要通过综合的、交叠的、充分的、全时的使用才能发挥出潜在的价值。移动计算终端正是这些资源最佳的使用者。

移动计算可以支持所有 Internet 应用,其特色应用是基于位置的服务。如果要提供个性化的信息服务,除了依据身份信息、兴趣爱好、作息时间、上网行为等因素外,移动计算又增添了所在位置这个重要的参量。而且,移动计算可以让使用者摆脱时间和空间的束缚。可以说,移动计算的优势是其能够最接近信息化的 5A 理想:任何人(Anybody)、于任何时候(Anytime)、在任何地方(Anywhere)、用任何方式(Anyway)能做任何事情(Anything)。

18.3.2 LBS

基于位置的服务(Location Based Service,LBS)是移动计算中较为重要和较有特色的应用。LBS 的基础是位置,即地理位置,包括经度、纬度、高度信息,有时也需要时间和速度信息。位置有准确和粗放之分,可以精确到数米,也可大致描述位于一座大厦、一个街区、一片区域乃至一座城市,不同的应用需要不同的精确度,但除了行车导航要求越精准越好,大部分应用其实并不需要达到很高的要求。

LBS 服务种类很多。例如,当移动终端查询城市道路交通、商业设施时,通过用户当前所在的位置,优先提供周边的信息;当用户出门在外使用移动终端时,如果正值午餐或晚餐时间,则可主动报告附近的餐饮店分布;当移动终端通过位置信息发现用户已经到达另一个城市或另一个国家,可自动显示当地的时间、当地的天气预报。

获取位置信息的方式有多种,应根据移动终端配置的硬件模块和不同的网络应用来选择和设计。

（1）GPS 定位精度高（可达数米以内）、专业性强，但存在初次定位时间长、运行能耗大、容易被建筑物遮挡等缺陷，无法在室内场所使用，通用性不够。因此，GPS 较少用于导航以外的应用。

（2）2G/3G 移动通信系统定位是通过手机与接入基站的相对位置来确定方位（类似三角定位原理），虽然定位精度大大低于 GPS，信号易受阻挡和反射的影响而加大偏差，误差可达百米以上，但手机（或带移动通信模块的其他手持终端）非常普及，而且全时在线，是一种比较实用的低精度定位方式。然而，基站定位的最大缺陷是必须依赖基站所属的运营商，单靠手机无法独立完成，因为手机并不了解基站所处的经纬度信息。

（3）WiFi 定位方式随着 WLAN 技术的普及而变得越来越有吸引力。几乎所有的智能手机、平板电脑、笔记本电脑等移动终端都具备 WLAN 通信接口，WLAN 基站（俗称 WiFi 热点）遍布家庭、校园、办公室和城市公共场所（无线城市），最妙的是绝大部分热点都是免费的。而一个 WLAN-AP 的有效覆盖范围为数十米，则根据接入点 AP 的 SSID 检索其位置信息，就可了解自己身在何处，精度和稳定度甚至高于手机基站定位。缺点是搜集并维护所有 WiFi 热点的位置信息是繁重而艰难的任务，除了当地政府有能力承担这一使命外，也许只有 Apple 和 Google 做得到。

（4）IP 地址定位以其采集信息简单、与硬件配置和接入方式无关的优势而具有很好的实用性，例如新闻网站可以根据用户的 IP 地址显示更多该用户所在省份和城市的消息。Internet 全网 IP 地址的分布情况相对固定，只需在每个区域采样少量 ISP 提供用户接入的 IP 地址，即可推算出 IP 地址网段和地理位置的关系。但是，如果接入点使用的是内网私有 IP 地址，则这一定位方式无效。

（5）BT 或 NFC 交换信息定位是基于互助式机制的信息共享方式。NFC 是继蓝牙（BT）之后出现的近距离（近场）无线通信技术（Near Field Communication），支持用户终端间点对点的数据交换。如果持有移动终端的人群中有一人掌握了当前位置信息，则可以通过 BT 或 NFC 将信息发布给周边人群。这一方法在应用中存在的问题是终端互连需要相互认证（以防止非法入侵），而且需要相关终端都开启 BT 或 NFC 功能。

关于 LBS 应用，人们一方面对其个性化、人性化、精细化的服务津津乐道，另一方面也对隐私保护问题心怀疑虑。这的确不是杞人忧天，因为位置信息是每个人最重要的私有信息之一，如果被人恶意利用，将严重侵害人身权利。

防范位置信息泄露、降低 LBS 风险是信息安全的重要研究内容之一。网络服务商、通信运营商的自律是最基本的要求（商业道德层面），从技术的角度看，应避免位置信息与身份信息的耦合度，实施原则是两者尽可能不进行关联计算，不在网络上一起传输。

LBS 应用有两种类型：一是正向利用位置信息，即显示终端当前所在的地理位置（如导航应用）；二是逆向利用位置信息，即根据终端所处的方位，显示周边或其他与位置相关的服务信息。大部分的 LBS 应用属于第二类。

18.3.3　App

当我们讨论信息系统的访问方式时，通常会围绕客户机/服务器（C/S）模型和浏览器/服务器（B/S）模型，然而，随着移动计算应用的展开，尤其是以 iOS 和 Android 为平台的智能移动终端的普及，人们开始接受一种创新的网络应用方式——App。

App是客户端应用程序(application)的简称。App在技术上就是一种客户机软件,但比传统意义的客户机轻巧、灵活得多。从C/S二层架构到B/S三层架构是一种进步。

一个App客户端软件同样面对服务端的后台系统,双方通过网络直接交换数据,是二层架构模式,但与C/S相比具有如下差别。

(1) App的服务端有两种:一是用于管理下载的软件库(例如AppStore),二是用于服务的信息源。App下载安装成功后,定期与软件库交换信息以更新版本(自动或手动),应用服务则由信息源提供,如上架新书、股票行情、即时消息等,一般需要启动App后才从服务端获取信息。

(2) App可以不依赖信息源而独立存在、独立运行,如计算器、电子游戏等;App也可以同样不依赖信息源而相互通信,与其他计算机设备通信,如电子名片交换、照片打印等。

(3) App具有功能单一、小巧灵活的特点,占用很少的资源,却可以根据自己的需要下载和组合,形成个性化程度很强的移动计算服务体系。

(4) 同样的应用可以有不同设计、不同实现的App,形成选择上的多样性,有利于发挥设计者的创造力,满足不同用户的需求。

(5) App的信息源不但是狭义的服务端系统,还可以拓展到整个Internet,从海量知识库中获得信息和服务;App客户端软件本身也不仅利用宿主机的计算能力,还可以利用移动计算终端的所有传感器信息,因此可以设计更丰富的应用功能。

(6) App是一种实现精准营销的广告载体。

采用云计算平台提供的标准化网络服务组件,基于移动计算终端开发丰富的App,是新一代网络信息系统的构建思路。

参 考 文 献

[1] 陈运.信息论与编码.北京：电子工业出版社，2009.
[2] 谢希仁.计算机网络.第4版.北京：电子工业出版社，2006.
[3] James D Solomon. Mobile IP(移动IP).北京：清华大学出版社，2000.
[4] Marcus Goncalves. IPv6网络.黄锡伟，等译.北京：人民邮电出版社，2000.
[5] S Rackley.无线网络技术原理与应用.吴怡译.北京：电子工业出版社，2008.
[6] 卢开澄.椭圆曲线密码算法导引.北京：清华大学出版社，2008.
[7] 蒋青.通信原理.第2版.北京：人民邮电出版社，2008.
[8] 唐宝民.通信网技术基础.北京：人民邮电出版社，2009.
[9] Jon W Mark.无线通信与网络.李锵译.北京：电子工业出版社，2006.
[10] 张玉艳.数字移动通信系统.北京：人民邮电出版社，2009.
[11] 徐明.移动计算技术.北京：清华大学出版社，2008.
[12] Michael Miller.云计算.姜进磊译.北京：机械工业出版社，2009.
[13] 丁贵广.多媒体技术.北京：机械工业出版社，2009.
[14] 王殊.无线传感器网络的理论与应用.北京：北京航空航天大学出版社，2007.
[15] A S Tanenbaum. Computer Networks.北京：清华大学出版社，2005.
[16] 斯托林斯.密码编码学与网络安全——原理与实践.第3版.刘玉珍译.北京：电子工业出版社，2006.
[17] 史蒂文斯.TCP/IP详解(卷1)协议.英文版.北京：机械工业出版社，2000.
[18] 张尧学.计算机网络与Internet教程.第2版.北京：清华大学出版社，2006.
[19] Douglas E Comer. The Internet Book(Internet技术基础).英文版.第4版.北京：机械工业出版社，2007.
[20] Jeff Doyle，Jennifer Carroll. Routing TCP/IP(TCP/IP路由技术).第1卷.第2版.葛建立译.北京：人民邮电出版社，2007.
[21] MaoZhen Li，Mark Baker.网格计算核心技术.王相林译.北京：清华大学出版社，2006.
[22] 麦勒瑞.Hardening Network Security.中文版.邓琦皓译.北京：清华大学出版社，2006.
[23] Forouzan B A. TCP/IP协议族.谢希仁译.北京：清华大学出版社，2006.
[24] 张友纯.计算机网络安全.武汉：华中科技大学出版社，2006.
[25] 郭军.网络管理.第2版.北京：北京邮电大学出版社，2005.
[26] 戴维森·福克斯.部署VoIP解决方案.凡璇译.北京：人民邮电出版社，2003.
[27] 孙利民.无线传感器网络.北京：清华大学出版社，2005.
[28] 强磊.基于软交换的下一代网络组网技术.北京：人民邮电出版社，2005.
[29] Forouzan B A.数据通信与网络.王嘉祯译.北京：机械工业出版社，2006.
[30] 黎连业.无线网络及其应用技术.北京：清华大学出版社，2004.
[31] 哈根(Hagen S)著.IPv6精髓.技桥译.北京：清华大学出版社，2004.
[32] 普兹马诺瓦.路由与交换.黄永峰译.北京：人民邮电出版社，2004.
[33] 李泽年.多媒体技术教程.史元春译.北京：机械工业出版社，2007.
[34] W Diffie，M E Hellman. New Directions in Cryptography. IEEE Trans Inform Theory，1976，IT-22/6：644～654.
[35] 徐国爱.网络安全.北京：北京邮电大学出版社，2007.

[36] 杨义先.网络安全理论与技术.北京:人民邮电出版社,2003.
[37] 凌力.网络协议与网络安全.北京:清华大学出版社,2007.
[38] 凌力.高级网络概论.北京:清华大学出版社,2011.
[39] 朱扬勇,凌力.客户/服务器数据库应用开发.上海:复旦大学出版社,1997.
[40] 曹文君,凌力.互联网应用理论与实践教程.北京:电子科技大学出版社,2001.